# SBSIVisual User Manual

Version 1.4.2

*Authors:* Richard Adams, Neil Hanlon,
Nikos Tsorman

January 11, 2012

# Contents


**1 Getting started** **4**
- 1.1 Starting the application . . . 5
- 1.2 Creating a workspace . . . 5
  - 1.2.1 Multiple workspaces . . . 6
- 1.3 Working with the example project . . . 6
  - 1.3.1 Installing the example workspace . . . 6
- 1.4 Creating a new project . . . 7
- 1.5 Hooking up to SBSINumerics . . . 8
- 1.6 New this version.. . . . 8
  - 1.6.1 New features in 1.4.2 . . . 8
  - 1.6.2 New features in 1.4.1 . . . 9
  - 1.6.3 New features in 1.4.0 . . . 10
  - 1.6.4 New features in 1.3.5 . . . 11
  - 1.6.5 New features in 1.3.4 . . . 11
  - 1.6.6 Bug-fixes . . . 12

**2 Running parameter optimisations** **13**
- 2.1 Run optimization locally . . . 14
  - 2.1.1 Viewing results . . . 15
  - 2.1.2 Common problems . . . 17
- 2.2 Choose data and Models . . . 18
- 2.3 Configure Parameters . . . 19
- 2.4 Configure optimization . . . 21
- 2.5 Configure Cost Functions . . . 21
- 2.6 Run optimisation remotely . . . 23
  - 2.6.1 Configuring a remote backend server . . . 24
  - 2.6.2 Cancelling jobs . . . 25
- 2.7 Test model compilation remotely . . . 26

**3 Running simulations** **27**
- 3.1 Run local simulations . . . 28

**4 Concepts** **31**
4.1 Model Configuration History . . . 32
4.2 The workbench . . . 32

**5 Miscellaneous Tasks** **34**
5.1 BioPEPA integration . . . 35
5.1.1 BioPEPA . . . 35
5.2 Creating a new SBSI project . . . 37
5.3 Importing and exporting projects . . . 37
5.4 Searching... . . . 40
5.5 Visualizing cost function history . . . 41
5.6 Visualizing optimisation results . . . 41
5.7 Visualizing a time series on a network diagram . . . 43
5.8 Visualizing time-series data . . . 45
5.8.1 Inbuilt data display . . . 45

**6 Reference** **48**
6.1 Miscellaneous features . . . 49
6.1.1 Reset perspective . . . 49
6.1.2 Preferences . . . 49
6.2 Problems view . . . 49
6.3 Progress View . . . 50
6.4 System view . . . 51
6.5 Installation test . . . 51
6.6 Updating SBSIVisual . . . 54
6.6.1 Updating SBSI . . . 54
6.6.2 Installing new software . . . 55

# Acknowledgements


## Funding

This work was supported by funding awarded to the Centre for Systems Biology at Edinburgh, a Centre for Integrative Systems Biology (CISB), by the BBSRC and EPSRC reference BB/D019621/1


## Contributors

SBSI is primarily the work of **Neil Hanlon, Nikos Tsorman, Shakir Ali, Azusa Yamaguchi, Allan Clarke and Richard Adams**. Others have contributed to this project, most of whom are listed below. If anyone who should be on is missing, we apologize, and please let us know so we can rectify the omission.
**Galina Lebedeva** provided example models and helped with testing the software.

**Andrew Millar** and members of the Circadian Clock modelling group provided direction, support and requirements to the project.

**Liz Elliot** and the CSBE admin team have provided a great deal of behind-the-scenes support to the project.

This project was led firstly by **Prof. Igor Goryanin** and currently by **Prof. Stephen Gilmore**.

For up-to-date information about help and contacts, please see our website:

`http://www.sbsi.ed.ac.uk`


**Copyright Centre for Systems Biology Edinburgh, 2010.**

# Chapter 1

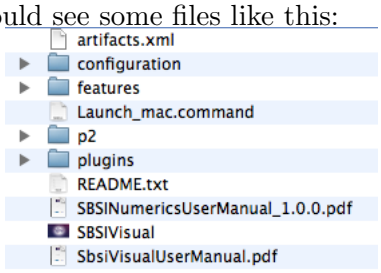
# Getting started

Welcome to this SBSIVisual user manual. This is a print version of the integrated Help pages contained within SBSIVisual itself. This documentation is valid for the 1.4.2 release of SBSIVisual.

## 1.1 Starting the application

Welcome to the Help pages for the SBSIVisual application. If you are a new user, you will have probably downloaded this application in order to run parameter optimizations of your biological model. This document describes the features of the SBSIVisual client application, but if you want more detailed information about the optimization procedure then please consult the SBSINumerics user manual that you can find in SBSIVisual's installation folder .

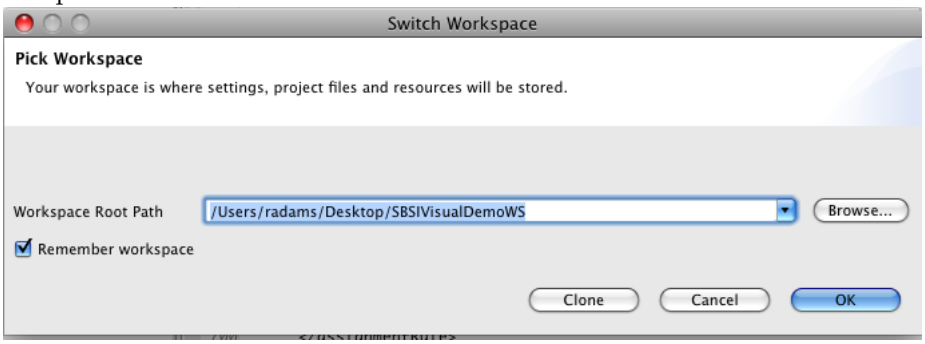
When you download and unzip the application, open the extracted folder and you should see some files like this:




The README file will have the platform specific information necessary to open the application.

## 1.2 Creating a workspace

The first time SBSIVisual is run you are presented with a dialog inviting you to choose a workspace.



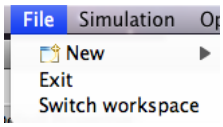

A workspace is just a folder that will hold your projects and resources. SBSIVisual listens to changes in workspace files and performs validation, error checking, and dynamic provision of functions based on changes. A good choice for the location of the workspace is :

- A writeable location in your own user space.
- Quite near to the file system root to avoid problems with file path lengths ( if using Windows ).
- Not easily modifiable by accident from outside the application.

The workspace files and resources should only be accessed from SBSI Visual - manipulating resources through the regular file system tools may result in the application not working properly.

### 1.2.1 Multiple workspaces

It is possible to have multiple workspaces on your filesystem. This is enabled by the **File->Switch Workspace** action.




which opens the workspace chooser dialog.

## 1.3 Working with the example project

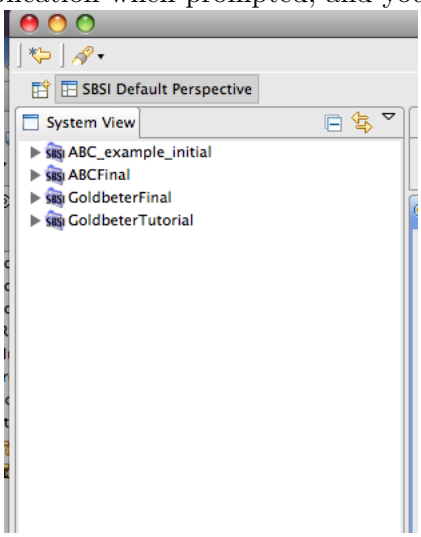
On Sourceforge, you can access an example workspace, containing example projects and results, to illustrate the sort of things you can do with SBSIVisual.

### 1.3.1 Installing the example workspace

Once you have SBSIVisual running, download and unzip the *SBSIVisualDemoWS.zip* workspace.

Now, in SBSIVisual, click: **File->Switch Workspace**

and choose the folder *SBSIVisualDemoWS* that you just downloaded. Restart the application when prompted, and you should see the following projects imported:




In the *Models* subfolder of the ABC_example_initial project, you can find an example SBML model representing the simple reaction system:

```
A -$>$ B -$>$ C
```

From here, you can run a simulation.

In the **Data->Experiments** folder, there are example time-series data sets in SBSI Data format . You can visualize a basic plot of the data, or see an animation of the data on a pathway diagram.

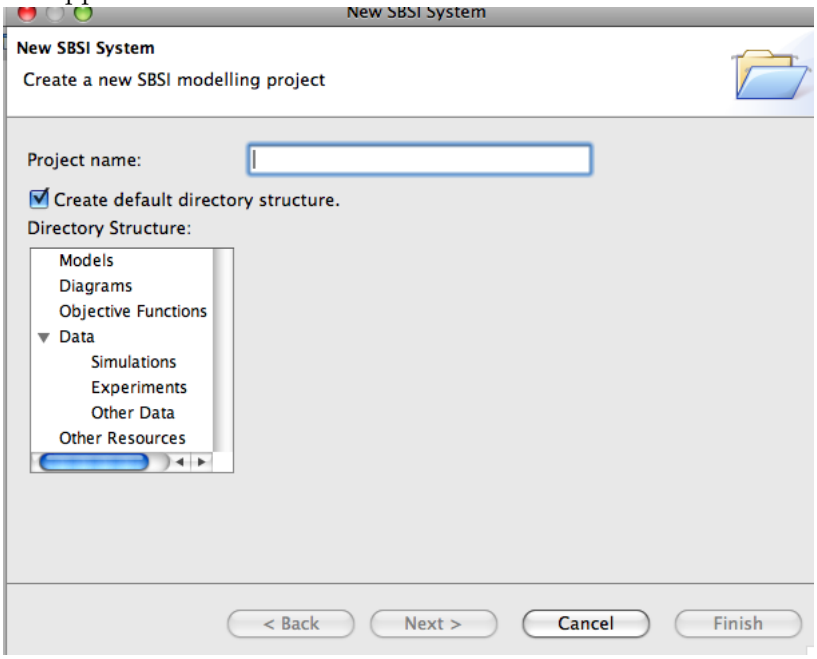
In the **Job Folder**, you can see the results of a pre-run optimization. The Help pages describing viewing optimization results provide links to how to visualize some of the result files.

If you have any question, please contact the SBSI team.

## 1.4 Creating a new project

When you run SBSIVisual for the first time, you will find an empty workspace. The first step you will probably take is to create a new SBSI system. This is essentially a specialized folder on your file system in which data, models and simulations can be managed.

To create a new Project, click **File->New->SBSI System**. The following dialog should appear:



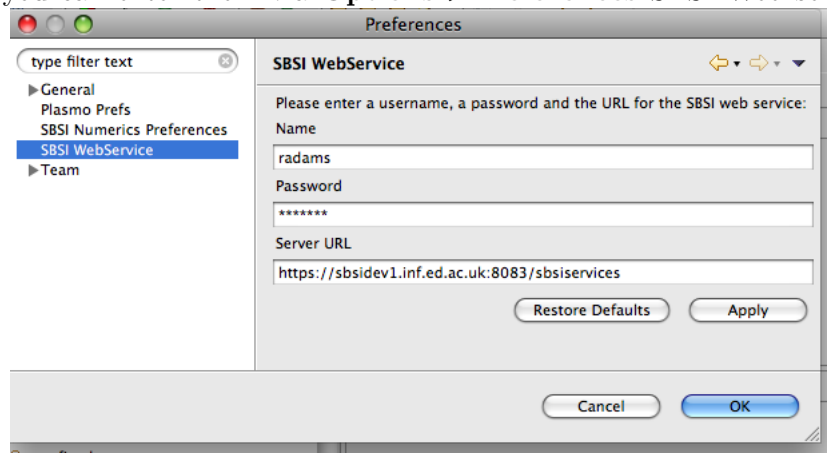

Enter a project name, which should be alpha-numeric. By default, a folder structure is created. If this folder structure is chosen, SBSIVisual uses this to help manage relationships between files. However you are entirely free to arrange your folders as you see fit. You can manipulate files and folders using standard methods, see the Workbench introduction for more details.

## 1.5 Hooking up to SBSINumerics

There are two ways to access SBSINumerics from SBSIVisual, in order to run parameter optimizations. Firstly, you can install SBSINumerics locally, or secondly, you can run optimizations on our server. (Or you can do both).

If you have installed SBSINumerics on your machine, you need to tell SBSIVisual where you have installed it. To do this, click **Options->Preferences** and then choose **SBSIVisual->SBSINumerics** preferences from the tree of possible preferences. Now, enter the path to your installation. For example, if you have installed SBSINumerics in /myhomeaccount/numerics, then that is the path to enter - it needs to be a full path from your file system. If you now try **Optimize->Local optimization** then you should be able to proceed with the configuration.

To do this, first of all please contact us to request a login and passwd. Once you have these credentials, you can enter them via **Options->Preferences** SBSI Web service server preferences.




You can click "Test Connection" to validate your credentials.

Once you've done this, you should be good to go - this is a one-time operation and SBSIVisual will remember what you've set.

## 1.6 New this version..

### 1.6.1 New features in 1.4.2

Tracker numbers for features and bugfixes are included in parentheses.

- SBSIVisual new features
  - Viewing optimization results - when downloading results, you can opt to view an HTML report summarising the run, which will open in a browser (#469). You can see this report subsequently by selecting the file 'ofwconfig.xml' in the sub-folder 'config_data' in your folder and choosing the 'Show Summary In Browser' action

- SED-ML files can now be viewed in HTML by selecting a SED-ML file, then selecting the 'Show in Browser' menu option.
- Data view tabs are now labelled with the data-set's underlying file name, rather than the generic 'Data View'. Also, by default all species are selected, and the chart's title is derived from the data file's name. (#474)
- Data views that are visualised by the 'Visualize Data' menu option can now be exported to SVG format via the context menu 'Export to SVG'.
- The FFT cost function can now be configured with an end time, after which the FFT will not be fitted.

### 1.6.2 New features in 1.4.1

Tracker numbers for features and bugfixes are included in parentheses.

- SBSIVisual new features
  - An SBML Validation Preference page is added - you can now choose what errors and warnings you want to see listed for your SBML file (#460).
  - When running a parameter fitting on the SBSI server, you can choose whether or not to compress the log files of the search, making the download shorter. (#459)
- SBSIVisual bugfixes
  - Testing model compilation can now proceed even if there are SBML errors, as these may be unimportant for SBSINumerics. (#460)
  - New SBSIVisual users can now access a simulation menu without an SBSI login. (#464)
  - Dialog history for TestModelCompilation added (#463)
  - Current Cost Function text-box resized to view the results better.
- SBSIDispatcher
  - To cut down on the size of the downloaded job contents, only time series data sets from the the last 1-2 checkpoints will now be returned.
- SBSINumerics bugfixes
  - SBSINumerics now ignores SBML errors pertaining to unit consistency, by default. It should not now fail if these errors are present in an SBML model.

### 1.6.3 New features in 1.4.0

Tracker numbers for features and bugfixes are included in parentheses.

- SBSINumerics
  - Particle swarm algorithms can now run multiple loops ( previously they could only be run in one loop) (#436).
  - A new configuration parameter, *ReportInterval* is introduced which specifies the interval between time-points in the 'Best time series' result files. This will limit the size of the output files (#303).
  - Negative values now accepted for the cost function *t_init* argument (#392).
- SBSIVisual new features
  - Upgrade of Copasi API library to 4.7 ( includes event handling )
  - Persistence of data animation layouts (#384)
  - Models and data can now be obfuscated on SBSI servers to improve privacy.
  - Performance of parameter constraint table greatly improved (#445).
  - Optimization jobs update automatically in the Dispatcher Database view.
  - Potential for smaller, and hence faster downloads of result files, by specifying larger intervals for the report interval (#303).
  - Improved feedback during the result download process (#438).
  - Model files in the result folders containing modified parameters now change their ID to contain the job name in which they were run (#441).
  - When configuring parameter constraints, default ranges for constraints can be applied globally to all parameters (#445). Previously, defaults were fixed at 0 (min) and 100 * the current value ( max).
  - The CVODE simulator used by the Numerics optimization framework is now available as a simulation provider via the Simulation menu ( requires network connection and an SBSI login ) ?(#327)
- SBSIVisual bug fixes
  - Downloading a job where no results were available does not now block subsequent downloads( #423).
  - Windows users can now download log files from the Dispatcher server (#437).
  - When configuring an FFT cost function, variables that are not selected are now not verified and do not block the user progressing through the submission wizard (#442).
  - Data file names containing spaces are now reported to the user during configuration (#444).

- Developer interest
  - SBSIVisual updated to run using Eclipse 3.7 components (#432).
  - Plugin extension point created for adding new simulation algorithms.

### 1.6.4 New features in 1.3.5

Version 1.3.5 is an incremental update with only one substantial new feature since 1.3.4 - the ability to weight the cost function values. See Cost function help pages for details, and section 6.7 of the SBSINumerics manual.

### 1.6.5 New features in 1.3.4

- SBSINumerics
  - Access to a parallelized Particle Swarm algorithm, an alternative approach to the genetic algorithm for parameter optimization. See section 6.6.2 of the SBSINumerics user manual for more details.
  - SBSINumerics now supports SBML models with species declarations containing **initialAssignment** elements.
  - Easier, script based installation on Unix-based machines.
  - Improvements to FFT cost function configuration - see section 6.7.2 of SBSINumerics user manual. You can now choose whether to factorize the cost by a function of the amplitude, or not.
- SBSIVisual
  - Access to SBSINumerics through a RESTful web services interface.
  - Simplified, more secure access to the SBSI server for running remote parameter optimizations.
  - You can now access the progress of your parameter fitting jobs on the web at https://mook.inf.ed.ac.uk:8083/sbsiservices/.
  - A new graphing and charting feature, which is more performant for plotting large data sets and has improved zooming and scrolling features.
  - SED-ML support is now fully compliant with the Level 1 version 1 specification.
  - Best parameters are now inserted into the model after optimization.
  - Standard menu items added : **File->Save**, **File->Save As** and **Edit**
- New plugins available for SBSIVisual
  - Access to the rule-based modelling framework Kappa via the SBSIVisual update site - see the SBSIVisual website for details.

– A graphical SBML editor, *VisualBiology*, is now available from the SBSI update site. Written by Dimitrios Milios at CSBE, this plugin provides a viewer and editor for SBML files.

Don't forget you can also get information from our Sourceforge Help Tracker at **https://sourceforge.net/projects/sbsi/forums/forum/1048774** which provides solutions and work-arounds for current usage issues.

### 1.6.6 Bug-fixes

- The inability of Windows versions of SBSIVisual to self-update has been fixed.
- More informative error messages should now be generated in the event of any problems importing the SBML model on the server.

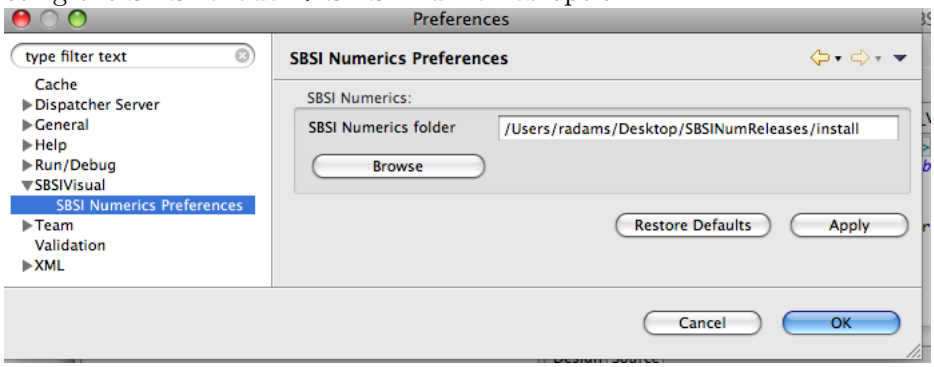
# Chapter 2

# Running parameter optimisations

## 2.1 Run optimization locally

In order to run an optimization locally on your own machine, the SBSI Numerics framework and its dependent libraries need to be installed. Instructions for this can be found in the SBSINumerics user manual.

Once it is installed, you need to point SBSIVisual to the install location of SBSINumerics. You can do this by selecting the **Options->Preferences** menu and then selecting the **SBSIVisual->SBSINumerics** option.




So, for example, if you installed SBSINumerics into **/home/sbsi**, then that is the folder you would choose in the Preferences.

Running local optimizations can be activated by or clicking on the **Optimize->Local optimization menu** To configure an optimization you need to :

1. Provide a model file from the workspace, in SBML format.
2. Provide one or more experimental data files and header files, in SBSI data format. See chapter 4 of the SBSINumerics user manual for details of this data format.
3. Configure the initial parameter values, which parameters to optimize, and constraints.
4. Provide a location in which the optimization can be run.
5. Configure the optimization process.

A wizard takes you through these steps and ensures all files are in the correct format before submitting the job to be run. See the tasks :

- Choose data and models
- Configure parameters
- Configure optimization
- Configure cost functions
- Choose job output folder

for guidance on using the wizard pages. The Problems View can help you resolve problems with data formatting.

You can view the progress of an optimization job in the Progress View. This shows the current best cost function value, and an indication of how long the job will run for if it runs until the 'MAX_NUM_GEN' cut off point is reached.

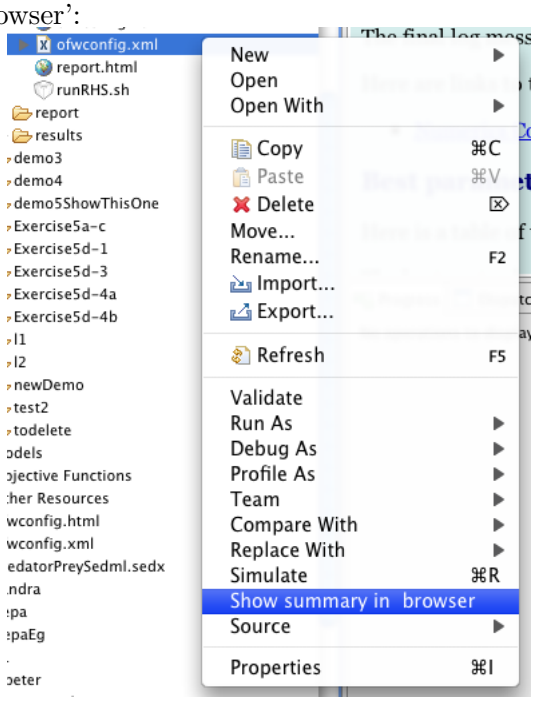
After a successful run, many files are present in the folder in which the optimization was run. The results can be found in a subdirectory,

./runtimeFolder/results/results and include several files:

- A time course of a simulation with the best found parameters
- A file containing the values of parameters which give the lowest cost
- A run log.
- A modified version of the model with the new parameters.

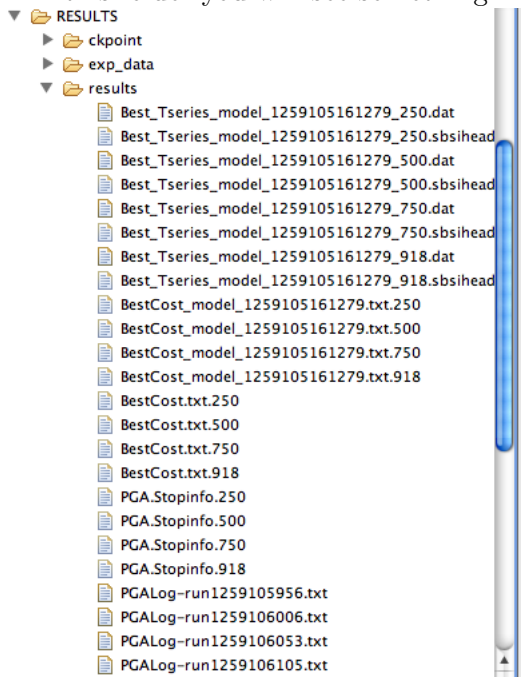
### 2.1.1 Viewing results

All the resources, results and configuration data from an optimization run are stored in a single folder. To get an overview of the results, go into the **config_data** subfolder and select the file **ofwconfig.xml**. From the context menu choose 'Show Summary In Browser':

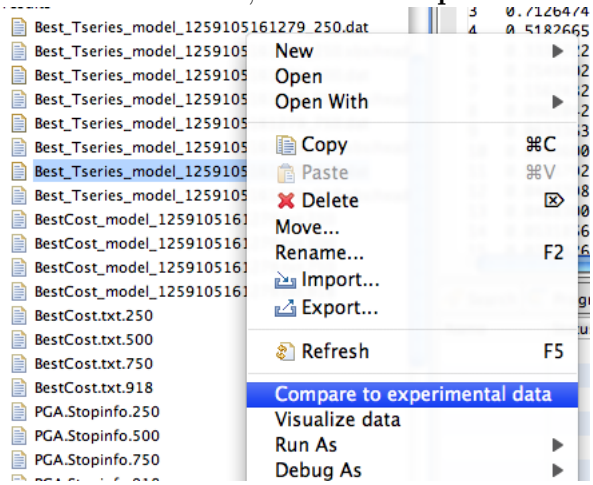



and an HTML view will open. If the optimization run produced results, you will be able to see thumbnail graphs of simulation results, the parameter values found, and links to the configuration files.

To explore the results in more detail, open up the job folder and open results/results folder. In this folder you will see something like the following:




Files ending in .dat suffix are the time series generated with the best parameters, at checkpoint intervals. To compare this with experimental data, select a data file and from the context menu, choose **Compare to Experimental Data**.

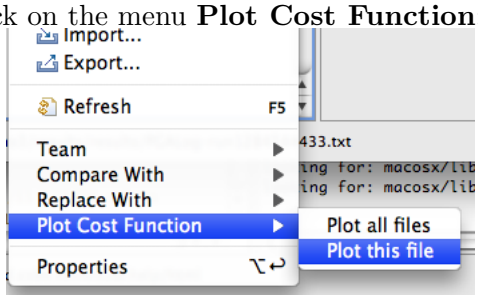



Then, from the ensuing dialog you can choose the data points and time series you want to compare.

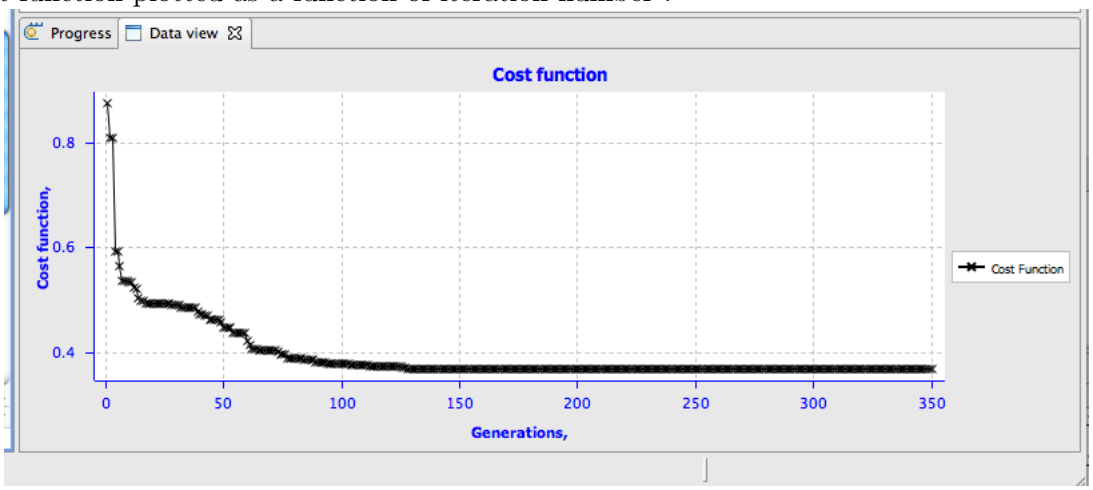
Alternatively, you can just visualize a single time series by selecting the data file and choosing : **Visualize data** from the context menu. Again, the dialog gives you the choice to choose which data sets to display, along with axes and titles. Another alternative is to **Quick Visualize Results**, which enables you to cycle through plots

of single species against a particular data set.

You can also inspect the overall progress of the optimization run by viewing a plot of the cost function value over time. To do this, access the results/ subfolder of the folder you selected for the output results, select a log file ( starting with PGALog.. ) then click on the menu **Plot Cost Function**:




You can choose to plot just a single file or all files - if there is ¿ 1 log file, they are listed in temporal order. Whichever choice you make, you will see a graph appear of the cost function plotted as a function of iteration number :




A slope with a negative gradient implies a decrease in the cost function over the search iterations ( the desired outcome ). A 'flatline' indicates that the search has become stuck in a local minima, or that the cost function landscape is 'flat' and there are many solutions with similar cost. In this case, you need to re-run the optimisation with larger mutation probabilites or a larger population size, to explore more of the parameter space.

Help pages for these tasks are available, e.g., Quick visualize, Visualize data

### 2.1.2 Common problems

There are several common problems :

- A model must have an *ID*. I.e., in the top level *model* element, there needs to be an id attribute. This is crucial for many aspects of the optimization process.

- For a parallel genetic algorithm, at least 3 parameters need to be chosen for optimization. This is necessary to allow the recombination of parameter values at each generation
- Some problems can be caused by choice of wrong scaling type for Chi-squared cost function. For example, if optimization 'hangs' with no sign of progress, this can often be overcome by reconfiguring scale type to 'Direct'.
- A job hangs, and attempted cancellation has no effect. This occurs sometimes if the job hangs during an initial 'set-up' phase. In this case, open a terminal window and type 'top'. This lists active applications, one of which will be 'modelID.exe'., where modelID is the SBML ID of your model. Note the process ID of the application, then quit top (Ctrl-C on Unix) and type:

  **kill -9 ProcessID**

  where ProcessID is the number noted from top. Type 'top' again and the process should no longer appear.

## 2.2 Choose data and Models

To run an optimization you must browse the local workspace and choose (in order) a project, a model in that project and finally dataset(s) in SBSI data format. You may choose 1 or more SBSI data sets. To select multiple data sets use **Ctrl-click** (Windows) or **cmnd-click** (Mac).

You will receive validation 'warning messages' (warnings do not block progress of the wizard and are displayed next to a yellow exclamation mark) if there is any inconsistency between the species in the data set and those in the model. You will also receive warnings if the chosen model contains SBML errors. Note, however, that the latter is very common and may not adversely affect the optimization.

You cannot choose 2 data sets with the same name. This creates an error, displayed next to a red cross and blocks progress of the wizard. Choosing a data set that does not have a valid header reference will also produce an error. Header references are found at the top of an SBSI data file in the format of

```
!/path/to/header.hdr
```

When they are invalid there will be a 'warning marker' - a yellow exclamation mark - in the margin of the file to the left of the header reference. If you enter a valid path to a header file and save the data file, the warning exclamation mark next to the previously invalid header reference will disappear. You may then return to the optimization wizard and proceed.

The file chooser wizard keeps a history of the chosen files for any given model. The history is saved at the moment the 'next' button is pressed and the wizard navigates to the next page. If a data file is moved from its historical location or deleted, then an

error message will be displayed until this file is removed from the selection or replaced by browsing to its current location in the workspace.

Note that when configuring cost functions the configuration is saved to the header files matching the data files chosen on this file browser page. Should you move the data files, but wish to keep the same cost function configuration for a given model, you will need to remember where in order to browse to them again.

Obviously, if you change the data files used, it follows that you will also use a different cost function configuration. This is in contrast to all other configuration of the optimization process which is linked solely to the model.

A selection list of optimization methods is present at the bottom of this page. It currently only includes PGA or simulated annealing - PGA is chosen by default. The choice made here affects the display and validation on the next page, the parameter configuration page.

## 2.3 Configure Parameters

Every parameter present in the model is displayed on this page in a 'checked table'. Each row in the table represents one parameter with a checkbox to toggle whether to use that parameter in optimization or not. Parameters are colour coded - parameters in light grey are not being used for optimization but may be toggled to being used. Parameters in dark grey are not being used and may not be toggled. This is because they are present in the experimental data. Parameters which are not 'constant' are toggled to 'off' by default and appear in light grey. A minimum of three parameters must be selected.

Parameters have constraints for max and min, an initial value and, if simulated annealing was chosen on the previous page, an initial step. All data on this page is saved to history as soon as it is edited. It is also validated at this time - invalid data is not saved to the history of the model. An attempt to enter invalid data generates an error message. Data is valid if:

```
min < initial value  < max (for PGA)
```

or

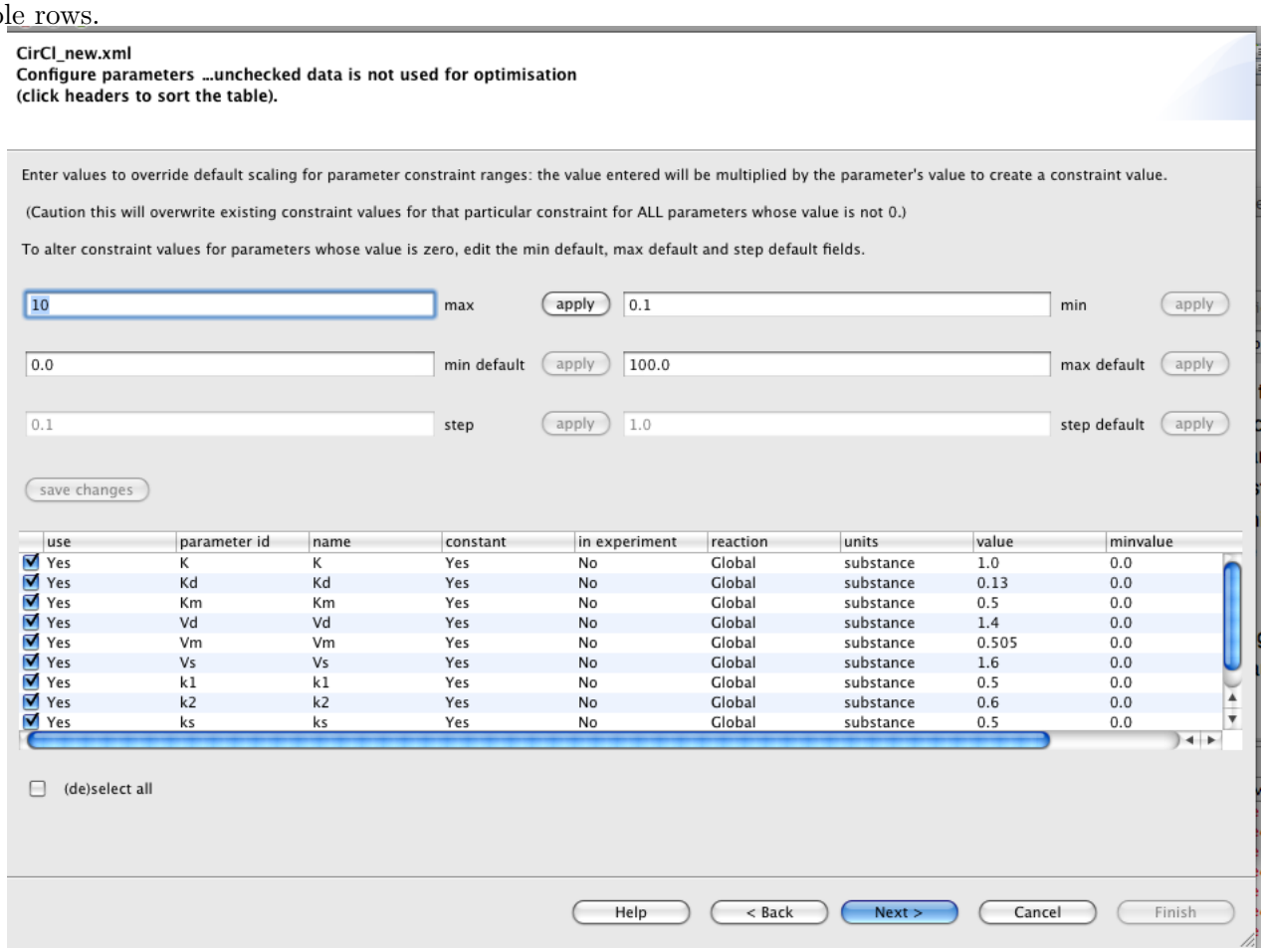
```
min < initial value + initial step < max (for simulated annealing).
```

Data can begin with a '-' and may have one instance of a '.' decimal point. Attempts to enter any other characters are ignored. However, it is possible to 'paste' in copied data that uses scientific notation. For example, its not possible to type 1.0E-10 for a value but it is possible to copy this string from a text editor and paste it into a cell in the table.

Note that parameter data from this page is saved in two places. One is the history of the model. The other is to the model itself. The reason that data is simply not just saved to the model, is that in SBML, we currently cannot save constraints to 'nested' parameters. Nested parameters are present inside reactions and are only visible within those reactions. (This is in contrast to global parameters). The existence of nested

parameters also explains the parameter names which are displayed on the parameter page in the wizard. Multiple parameters may have the same parameter ID - however, they will all be nested inside different reactions.

The parameter table is sortable by any of the column headers, including the 'use' column. Note that when sorted by the 'use' column, toggling parameters on and off will cause them to move position in the table, as toggling causes an instant re-sort of the table rows.


CirCl_new.xml
Configure parameters ...unchecked data is not used for optimisation
(click headers to sort the table).
Enter values to override default scaling for parameter constraint ranges: the value entered will be multiplied by the parameter's value to create a constraint value.
(Caution this will overwrite existing constraint values for that particular constraint for ALL parameters whose value is not 0.)
To alter constraint values for parameters whose value is zero, edit the min default, max default and step default fields.
10
max
apply
0.1
min
apply
0.0
min default
apply
100.0
max default
apply
0.1
step
apply
1.0
step default
apply
save changes
use
parameter id
name
constant
in experiment
reaction
units
value
minvalue
Yes K K Yes No Global substance 1.0 0.0
Yes Kd Kd Yes No Global substance 0.13 0.0
Yes Km Km Yes No Global substance 0.5 0.0
Yes Vd Vd Yes No Global substance 1.4 0.0
Yes Vm Vm Yes No Global substance 0.505 0.0
Yes Vs Vs Yes No Global substance 1.6 0.0
Yes k1 k1 Yes No Global substance 0.5 0.0
Yes k2 k2 Yes No Global substance 0.6 0.0
Yes ks ks Yes No Global substance 0.5 0.0
(de)select all
Help
< Back
Next >
Cancel
Finish

It's also possible to set the constraints globally, as a multiplier of the original value in the model. For example, to set the constraints to 90% and 110% of the parameters current value, type '1.1' into the 'max' textbox and '0.9' into the 'min' text box. Clicking 'Apply' will set these constraints globally.

This approach is useful when there are many parameters you wish to set constraints for at once.

## 2.4 Configure optimization

This page enables configuration of an optimization run. As this is a complex topic, please refer to the SBSI Numerics user manual for a detailed description. However, the page itself is designed to help and contains much of the relevant information describing each configurable item on it. Clicking on the 'information' symbol next to any text field spawns a pop up with help text. Text fields on the page are greyed out and disabled based on choices either made previously (i.e., PGA versus simulated annealing) or on the page itself (ie - use SobolSelect or SobolUnselect). If invalid data is entered an error is displayed. In contrast to the other wizard pages data is **not** saved until the 'save' button is pressed. No validation happens until this time either. Invalid data will be left in the text field awaiting correction. This is again in contrast to the other pages where invalid data will immediately be replaced with the original data upon attempting to finish an edit of the text field. If multiple text fields have become invalid then only 1 error message will be displayed. As soon as the field matching that message is fixed then the another error message will be displayed and so on until all invalid fields are fixed and the wizard can proceed to the next page.

The message 'save complete' is displayed when all errors on the page have been fixed. The 'Use defaults' button will revert the page to the default setting for each field and save these values at the same time.

There are two fields on the page, *tFinal* and *popsize* which depend on choices made on previous pages. Specifically, *tFinal* depends on the maximum time value of a data point in the data sets chosen for this optimization. Popsize is calculated as 50 multiplied by the number of processor (cores) on the platform which will run the optimization. (For remote optimizations the number of processor cores = the 'tasks' value configured in the remote server configuration section).

If the saved values in the history for these fields don't match the currently calculated values, an error message is displayed and the wizard is blocked until these values are confirmed by clicking on the 'saved'/'calc' choice buttons that appear. You may choose either to use the saved value or the newly calculated value. This will not need to be confirmed again unless you revisit the page which originated the calculated value (the parameter page for popsize or the file browsing page for tFinal).

## 2.5 Configure Cost Functions

The Cost Function page is split into two sections. The top section is for FFT configuration, the bottom is for chi-squared (X2). FFT also contains an adjacent table displaying states/species present in the model. It is important to note that when cost function configuration dialog history is saved, it is written to the SBSI header files. This is in contrast to the rest of the configuration for an optimization job, which is linked to the model. Therefore if you copy/move a model, its configuration history is copied or moved with it but the cost function configuration history does not - it is associated with the SBSI data files.

You can get the current cost of your model, prior to optimization, via the **Get Current Cost** button. You can therefore tell if fitting produces a better result than what you have already with your existing model values.

The Cost Function Page validates user input so that obviously incorrect data cannot be used. Error messages are displayed at the top of the page and progress in the wizard is disabled until inconsistencies in the data are resolved; you may have to use the previous job configuration page to do this as values here can conflict with values on the cost function page.

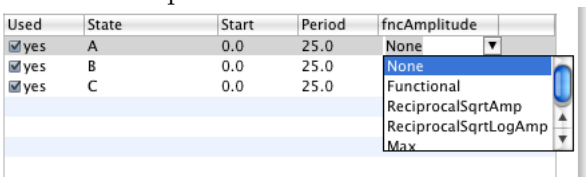
For FFT cost function, a warning is displayed to you if your start time plus 2* period (for a state) is less than the tFinal value set on the previous page. This is just intended as a helpful pointer that the framework will have little possibility of finding cycles.

For both FFT and X2 Cost functions, click on the **'use in setup'** checkbox in order to include or exclude them from the setup phase of optimization (although you must use at least 1 cost function in this phase). *Interval* must be greater than zero and greater than the solver interval defined on the job configuration page. The start time (t_init)for an X2Cost Function is actually an offset from the start time in the dataset. The default value, 0 means that the data time points should be compared with the equivalent simulation time points. An offset of -5 would compare data with simulation data at t -5. (The corresponding value, tFinal, determines the simulation end-point and set on the job configuration page as it is global for all cost functions).

The 'Weight' term is an optional configuration to scale the relative importance of X2 and FFT cost functions when both are selected. By default, the costs for chi-squared and FFT functions are simply summed - if there is a difference in scale between these values, then the smaller contribution will be masked. Increasing the weight for the smaller value would counter this effect. See section 6.7 of the SBSINumerics manual for a complete explanation.

For FFT cost function, the state data can have a start value and expected period set for each individual state in the model. Uncheck the 'use' checkbox to exclude the state from optimization.

From v1.3.5, we have introduced an option to configure the FFT cost value based on a function of amplitude:




The details of this are explained in the SBSINumerics user manual, section 6.7.2, but there are 3 options:

- None - the FFT cost is not scaled with respect to the amplitude of the peak and is therefore amplitude agnostic.
- Functional (synonym= ReciprocalSqrtAmp) - the FFT cost is scaled by 1 / sqrtamp where *amp* is the amplitude of the peak

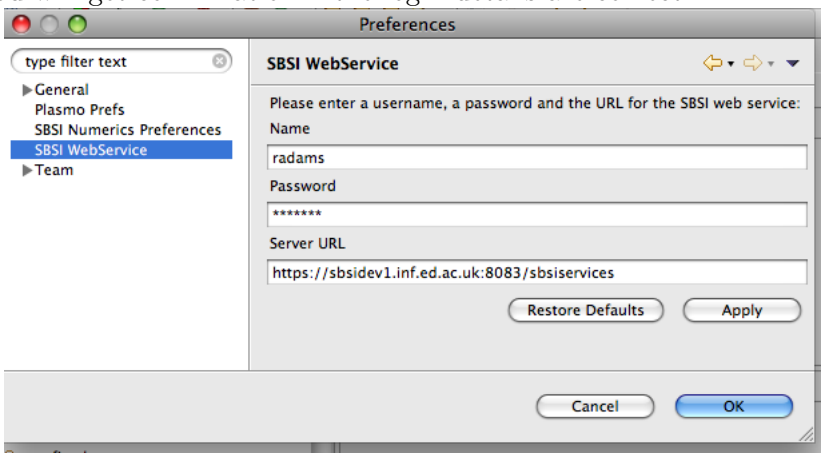
- Max(synonym=ReciprocalSqrtLogAmp) - the FFT cost is scaled by 1 / sqrt-log(amp).

.

## 2.6 Run optimisation remotely

In order to run an optimization remotely, no installation of the SBSINumerics optimization framework is required. However you will need to be registered to run on a remote machine. For example, if you want to run optimization on a remote supercomputer, you will need to have set up an account independently on that supercomputer. If you want to run jobs on our SBSI server, then please get in touch and we'll send you login details.

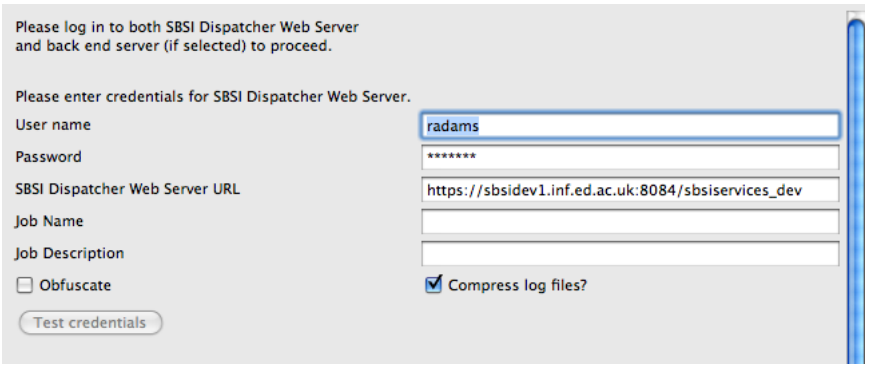
Once you have completed this first login page, configuring an optimization job is exactly the same as configuring it to run locally. To set up your login, click on: **Options->Preferences->SBSIWebService** and enter the login credentials. Then click 'Apply' - you will get confirmation if the login details are correct.:




Running remote optimizations can be activated by or clicking on the **Optimize->Remote optimization** menu.

The first wizard page asks you to give a name/description for your optimization, that you can use to identify it. Optionally, you can choose to obfuscate the model when it is sent to the server as an additional security step. This garbles all the identifiers in your model and data so that there will be no way to identify the biological context of the model while it is run on a backend server. When the results are subsequently downloaded, the files are deobfuscated on your machine to restore the original identifiers.

Another option is to compress the log files generated during the search. By default, this is enabled. This means that the full search history of all the tested parameter values is discarded, and only a record of the cost-function values is returned. This makes the downloads much smaller, especially for large fittings with many parameters.



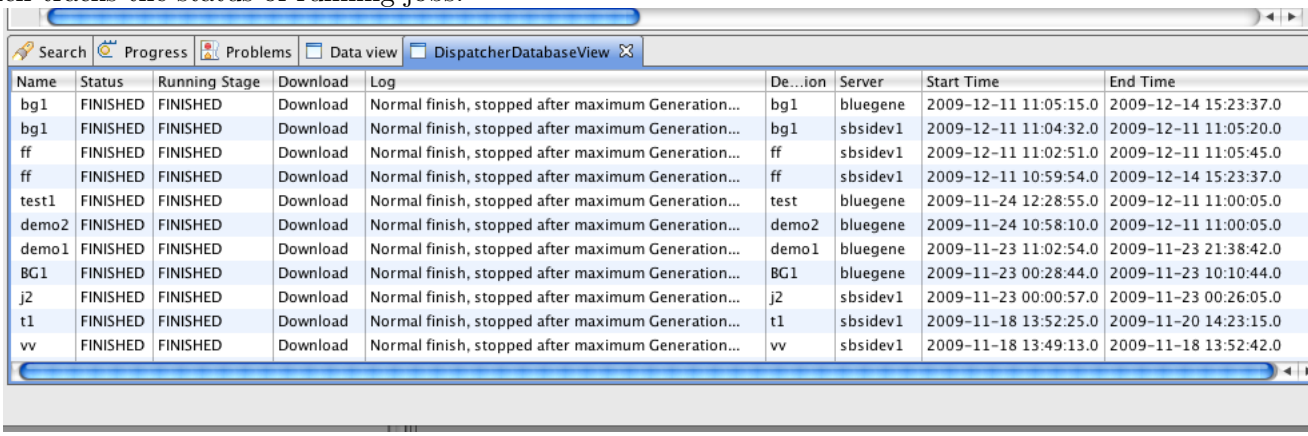

### 2.6.1 Configuring a remote backend server

This section of the login page is a generic login for a backend supercomputer. There are 4 fields to complete:

- The host name of the server.
- The number of processors to request. This will be determined by the size of your model and the characteristics of the backend server.
- Wallclock time - this is a time after which a job will be cancelled. Again, an appropriate setting will need to be found by trial and error, together with knowledge of the available settings on the backend server. Requesting a longer wallclock time may result in a longer wait in the supercomputer queue.
- Optional command line arguments - this is an optional field and its contents will be backend-machine specific.

Once you have completed the first login page, configuring an optimization job is exactly the same as configuring it to run locally.

You can view the progress of a remote optimization job by choosing the **Optimize->Query Database** menu item. This opens a new view, the Dispatcher Database view, which tracks the status of running jobs:




| Name | Status | Running Stage | Download | Log | De...ion | Server | Start Time | End Time |
|---|---|---|---|---|---|---|---|---|
| bg1 | FINISHED | FINISHED | Download | Normal finish, stopped after maximum Generation... | bg1 | bluegene | 2009-12-11 11:05:15.0 | 2009-12-14 15:23:37.0 |
| bg1 | FINISHED | FINISHED | Download | Normal finish, stopped after maximum Generation... | bg1 | sbsidev1 | 2009-12-11 11:04:32.0 | 2009-12-11 11:05:20.0 |
| ff | FINISHED | FINISHED | Download | Normal finish, stopped after maximum Generation... | ff | sbsidev1 | 2009-12-11 11:02:51.0 | 2009-12-11 11:05:45.0 |
| ff | FINISHED | FINISHED | Download | Normal finish, stopped after maximum Generation... | ff | sbsidev1 | 2009-12-11 10:59:54.0 | 2009-12-14 15:23:37.0 |
| test1 | FINISHED | FINISHED | Download | Normal finish, stopped after maximum Generation... | test | bluegene | 2009-11-24 12:28:55.0 | 2009-12-11 11:00:05.0 |
| demo2 | FINISHED | FINISHED | Download | Normal finish, stopped after maximum Generation... | demo2 | bluegene | 2009-11-24 10:58:10.0 | 2009-12-11 11:00:05.0 |
| demo1 | FINISHED | FINISHED | Download | Normal finish, stopped after maximum Generation... | demo1 | bluegene | 2009-11-23 11:02:54.0 | 2009-11-23 21:38:42.0 |
| BG1 | FINISHED | FINISHED | Download | Normal finish, stopped after maximum Generation... | BG1 | bluegene | 2009-11-23 00:28:44.0 | 2009-11-23 10:10:44.0 |
| j2 | FINISHED | FINISHED | Download | Normal finish, stopped after maximum Generation... | j2 | sbsidev1 | 2009-11-23 00:00:57.0 | 2009-11-23 00:26:05.0 |
| t1 | FINISHED | FINISHED | Download | Normal finish, stopped after maximum Generation... | t1 | sbsidev1 | 2009-11-18 13:52:25.0 | 2009-11-20 14:23:15.0 |
| vv | FINISHED | FINISHED | Download | Normal finish, stopped after maximum Generation... | vv | sbsidev1 | 2009-11-18 13:49:13.0 | 2009-11-18 13:52:42.0 |

Clicking on the column titles sorts the table by that column. When some results are available, it is possible to download them by clicking on the **download** button. You will then be prompted for a location in your workspace to save the results. You can click on 'Refresh' button in the top-right corner of the table to refresh the view of the progress.

Results can be viewed in a similar way to a locally-run optimisation. The table view shows job status in two ways - a 'coarse-grained' status ( one of Running, Finished or Error ) and a finer grained 'Running status' which tracks the progress of running jobs. The Running stages are as follows :

- *RECEIVED* - a new job has been submitted from the client. This is the initial state before any processing occurs.
- *VALIDATED* - basic validation of submitted resources is successful. I.e., this checks that all necessary files are available for the job
- *COMPILED* - the model has been converted to C ,and a test compilation was successful.
- *SUBMITTED_TO_REMOTE_SERVER* - the job has been submitted to the server that will run the job
- *LAUNCHED* - the job has now started executing
- *PARTIAL_RESULTS* - the job is still running but has produced some interim results.
- *FINISHED* - the job has stopped running, either successfully or with error.

From any point from '*SUBMITTED_TO_REMOTE_SERVER*' onwards, then some download should be available. Before the job is launched this may be useful to get some indication of an error, for example, or to examine log files. Once '*PARTIAL_RESULTS*' stage is reached, some optimisation results will be available. The frequency which this is updated depends on the configuration of the optimisation - every *numGen* generations an interim set of results (checkpoint) will be produced. The job is FINISHED when the optimisation has finished, perhaps because the target cost has been reached, or the MAXGEN number of generations has been reached.

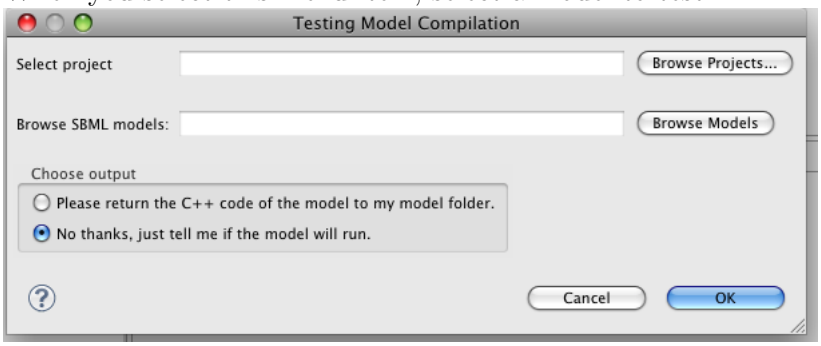
### 2.6.2 Cancelling jobs

Running Jobs can be cancelled by clicking on a cell in the 'Status' column, which enables a 'Cancel' button. This will stop the remote job, and update the running stage to be correct at the time of cancellation. If results are available at the time of cancellation, these can still be obtained by the 'Download' mechanism described above.

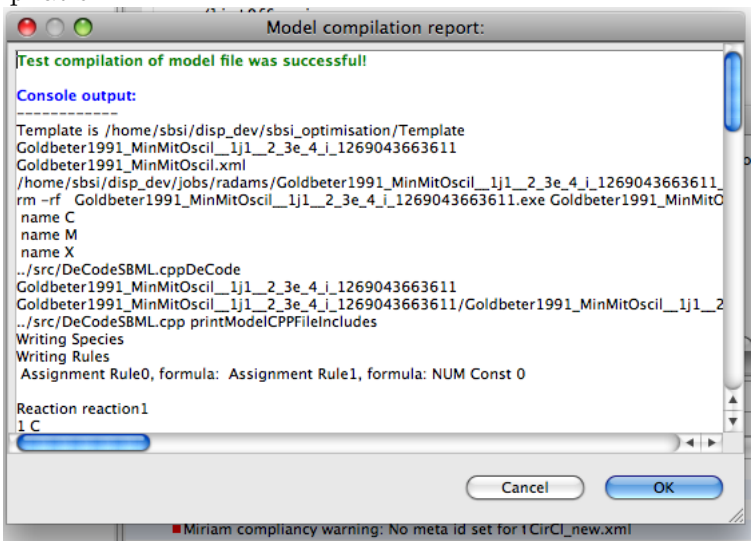
## 2.7 Test model compilation remotely

Before configuring an optimization job it is a good idea to check that SBSI will be likely to run your model. Using the **Optimize->Test Model Compilation** menu item it is possible to do this.

When you select this menu item, select a model to test.




If you like, you can also download the generated C++ code. This could be useful if you are interested in developing your own custom cost functions for your model. After running, a dialog should appear, reporting on the success or otherwise of the test compilation.




If successful, it is likely your model is suitable for optimising with SBSI. If not, then please send your error report to the SBSI development team for advice.

# Chapter 3

# Running simulations

## 3.1 Run local simulations

Running local simulations can be activated by **Ctrl +R**, by clicking on the **Simulation** main menu, or by selecting an SBML file in the System View and accessing the Simulation command from the context menu. To configure a simulation you need to :

- Provide a model from the workspace.
- Optionally specify a location to save the results to.
- Choose a simulation algorithm.
- Optionally configure the simulation parameters.
- Optionally configure how you want the results to be presented.

The simulation algorithms are provided by third party libraries, and SBSI does not itself check that the chosen algorithm is appropriate for the model. In general, the deterministic LSODA algorithm from COPASI works reliably for curated models from the Biomodels DB.

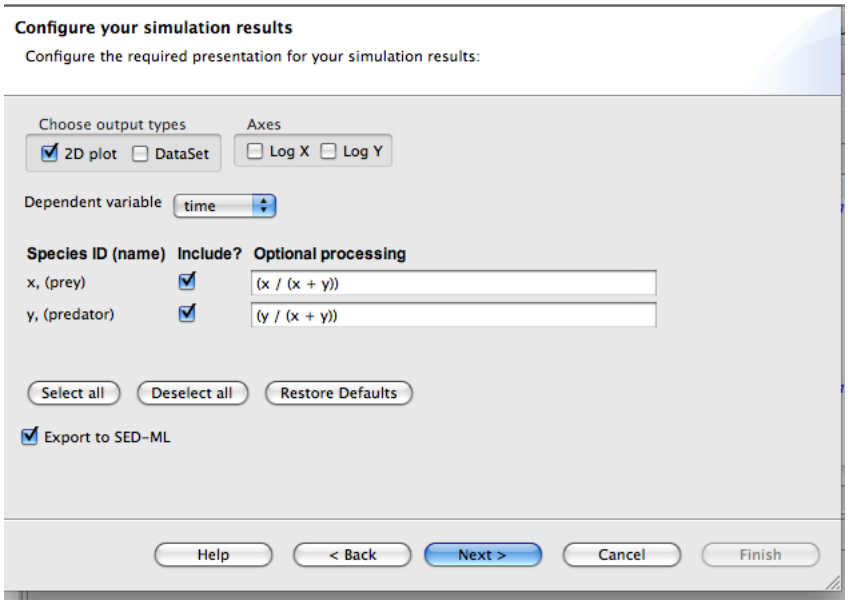
The following algorithms are bundled with SBSI:

- Copasi - LSODA, Stochastic (hybrid)
- Dizzy - Runge-Kutte (level 1 SBML models only)

If you have SBSI-Numerics, then you can access its inbuilt CVODE solver by setting the location of your install in **Options->Preferences->** and then select **SBSIVisual->SBSINumerics** preferences. Alternatively, you can access the CVODE solver via a web service. This is useful to run prior to submitting an optimization job, since it will confirm whether or not SBSI can simulate your model properly.

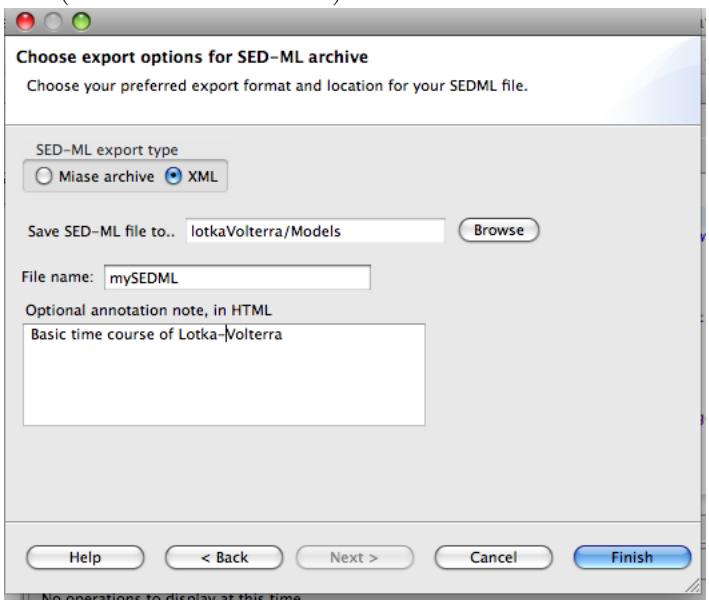
For a fully featured simulation tool that handles stochastic and ODE based models, you can install the Biopepa plugin.

In the output configuration page, you can choose log axes, whether you want to save the simulation data in a file, and which variables you want plotted. You can also choose to post-process the simulation results by entering a mathematical expression.




If you choose to save the data, and want to look at it again, you can use the Visualize data menu item to do this.

If you want store your simulation configuration, for archival purposes, or for running in another software tool, you can export your simulation configuration in SED-ML format. To do this, you just need to choose an export file and location, and whether you want to export just the description, or include the model as well in a self-contained archive ( a 'SED-ML archive').




If you do the former, you will need to edit the generated XML file to add in an appropriate URI for the 'source' field of the model element. If you export the archive, the model will be included. The archive is in 'zip' format but with an extension '.sedx'

and you can access the original files in the future by unzipping the archive.

A history of simulation runs, and the ability to monitor the progress of running simulations, is available in the Progress view.

SED-ML files can be used as a way to run simulations as well, and SBSIVisual offers some support for doing this. To do this, right-click on a SED-ML file and choose the menu 'Execute SED-ML simulation'. A dialog will appear with a choice of *Outputs* described in the file, for you to reproduce. At present in SBSIVisual we do not support 3D visualizations, so only 2d plots and reports (tables of data) can be generated.

You can also view the contents of a SED-ML file by selecting it and choosing the 'View in Browser' menu item which will open an HTML view of the SED-ML file.

Once an output is chosen, SBSIVisual will attempt to run the simulation. The success of this will depend on many factors:

- The model language. Currently SBSI will only simulate SBML models.
- The simulation type, as described in the 'Algorithm' element of the SED-ML file.
- Whether the model file can be found. In SED-ML, models are referenced by a 'source' attribute in the SEDML *model* element. Currently, the actual model file needs to be on your computer, and the value of the source attribute should be either an absolute path or a relative path to the model file, relative to the SEDML file. In future we hope to provide support for resolving models from common model repositories, using web services.

If you choose to reproduce a *Plot2D* output, a graph should appear. If you choose to reproduce a *Report* output, you will be prompted for a file location to save the report. Any problems or failures will be logged and displayed.

# Chapter 4

# Concepts

## 4.1 Model Configuration History

The model configuration history includes all of the necessary information to repeat a model's last optimization run. Furthermore any given optimization job can be re-run using the precise configuration which it originally used. The mechanism is designed to maintain this history if you copy/move or delete a model, whether SBSIVisual is running at the time or not. Models which are copied will get a copy of the configuration history of the original. They will not subsequently overwrite the original's configuration.

1. To re-run the last optimization job of a model, just launch the wizard with that model. All configuration parameters are the same (including cost function, if none of the data sets have been moved - see below).

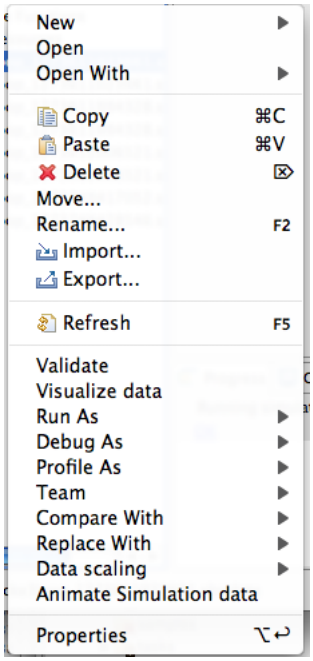
2. To re-run a given optimization job, run the wizard choosing the model inside the 'config_data' folder of that job's output. Also choose the data sets inside the config_data folder (choose all of them, as they were all used in the original run).

However there is a caveat to the use of the configuration history:

1. Please note that cost function configuration is linked to the data sets being used and not to the model. A history is kept of which data sets were used for an optimization job but if they are moved then this is lost.

## 4.2 The workbench

The workbench is a folder on your filesystem that the application uses to manage files. The application can respond to changes to files in this directory, and run processes such as format validation in the background. It is best if the user does not modify this filesystem outside of the application. By default, the workspace is set up in the installation directory of the application. The System view provides a tree-view onto the filesystem, and allows typical operations such as opening files, deleting , renaming, etc., available from the context menu:


New
Open
Open With
Copy ⌘C
Paste ⌘V
Delete
Move...
Rename... F2
Import...
Export...
Refresh F5
Validate
Visualize data
Run As
Debug As
Profile As
Team
Compare With
Replace With
Data scaling
Animate Simulation data
Properties ⌥↩

# Chapter 5

# Miscellaneous Tasks

## 5.1 BioPEPA integration

### 5.1.1 BioPEPA

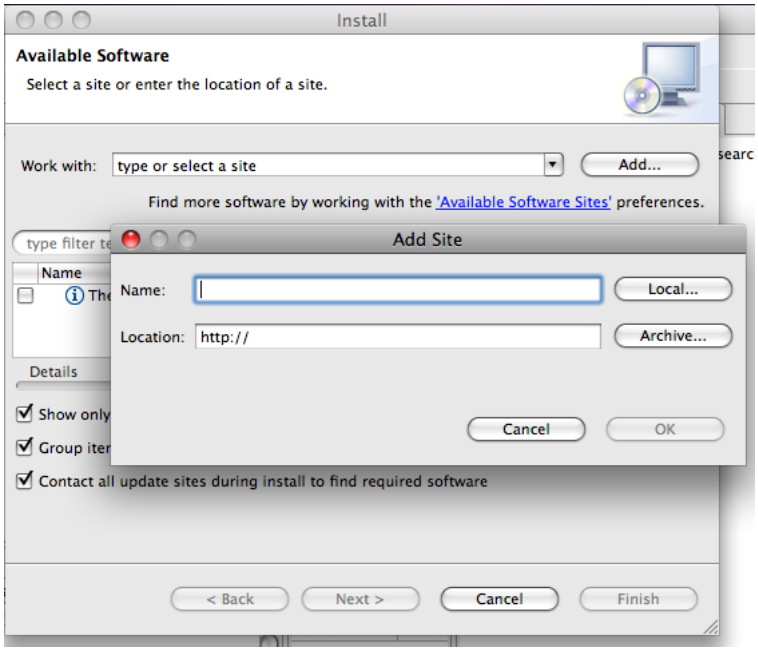
BioPEPA (http://www.biopepa.org) is a stochastic modelling framework based on process algebras. It is an independent software effort, but can be run within SBSIVisual. SBSIVisual may have been distributed with or without BioPEPA. This page describes how to install BioPEPA if you don't already have it. If you're not sure if you have BioPEPA installed or not, click **Window->Show Perspective** - if the BioPEPA perspective is listed, then you have BioPEPA installed and can skip this section.

The installation process uses SBSIVisual's update facility which is a mechanism to install new functionality in a straightforward manner. You will need to be networked in order to download the BioPEPA plugins.

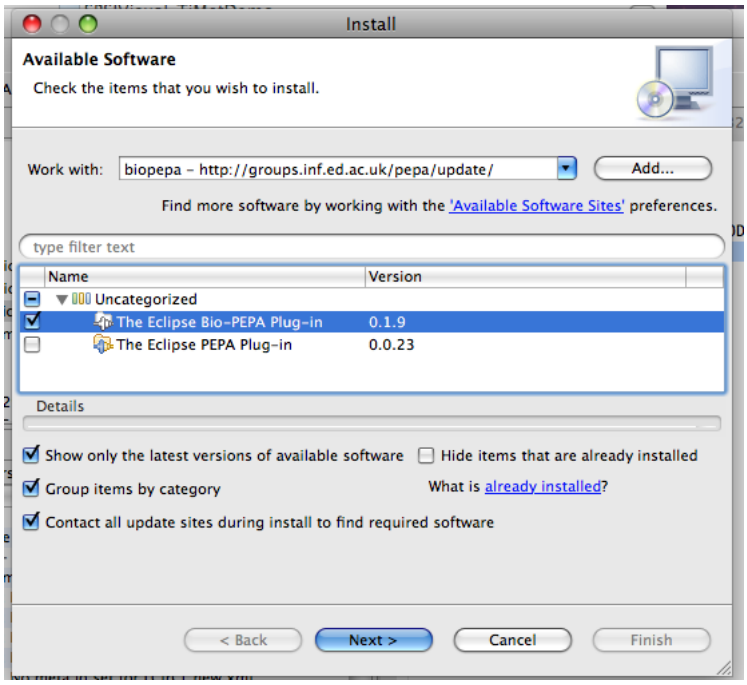
New features can be downloaded and installed from *update sites*. It may be that your installation already knows about the update sites it needs to contact - to test this, Click **Help->Install New Software** and expand the **Work with** dropdown list and see if the BioPEPA update site is listed. If it is, you can skip this first section. If not, we need to add in the URLs of the BioPEPA update site, and two other update sites on which it depends:

To install BioPEPA, the following steps are necessary.

1. Click **Help->Install New Software**. In the resulting dialog, click on the 'Add' button and add the following names and urls.
    - Name: BioPEPA, Location: http://groups.inf.ed.ac.uk/pepa/update/
    - Name: BIRT, Location: http://download.eclipse.org/birt/update-site/2.5
    - Name: emf, Location: http://download.eclipse.org/modeling/emf/updates/




2. Click **Next**, then in the 'Work With' drop down list choose 'Eclipse BioPEPA plugin', keeping other defaults the same. ¡span style="font-weight:bold"¿ It is essential that the check box 'Contact all update sites during install' is selected for the installation to succeed¡/span¿.




3. Click **Finish**, accept any prompts for licence agreements, then install. Installation

will download about 50Mb of new code. After this, accept the invitation to restart.

In the restarted application, the BioPEPA perspective should be enabled in the **Window->Show Perspective** view. For help with using BioPEPA, see the project webpages at www.biopepa.org

## 5.2 Creating a new SBSI project

You can create a new project by clicking **File->New->SBSI System** or **Ctrl-N -> SBSI System**. A wizard dialog pops up, which needs an alphanumeric name for the project. You can also choose whether you want the default project directory structure set up.

In the next page of the dialog, you can optionally import an SBML file into your file system. Before importing, SBSI attempts to validate the file and shows a validation report.

On finish, the resources are created in the workspace and made visible in the System view.

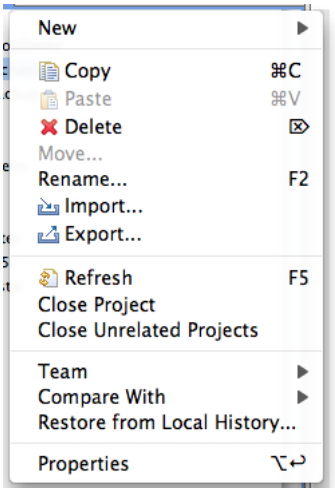
## 5.3 Importing and exporting projects

This page tells you how to import and export projects into and out of SBSIVisual. Files and folders can be dragged and dropped into the workspace. Projects on the other hand can be imported and exported via a dialog.

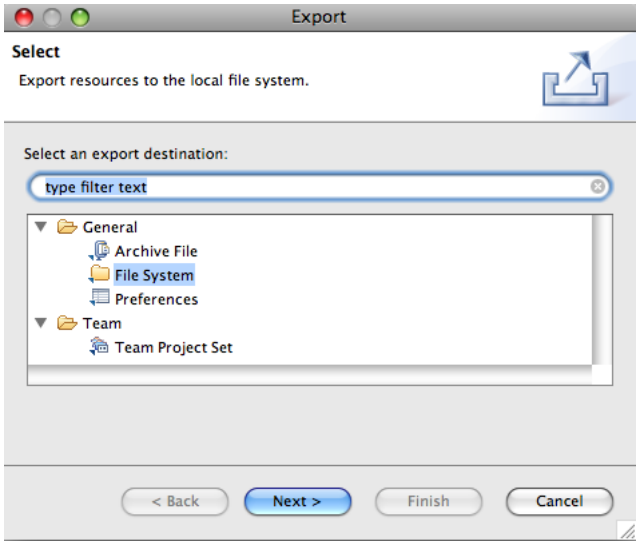
- Exporting projects
- Importing projects

To export a project, to share with a colleague, for example:

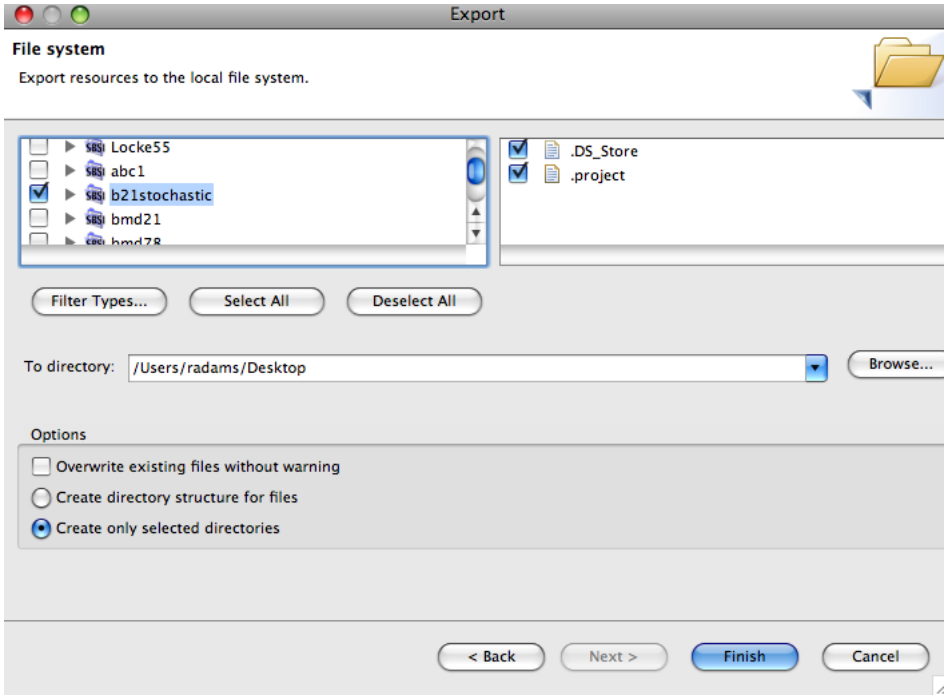
1. First of all select the project you want to export in the System View, and right click to get the context menu - choose the **Export** menu item:



2. In the subsequent dialog, choose export to File System, then click **Next**:

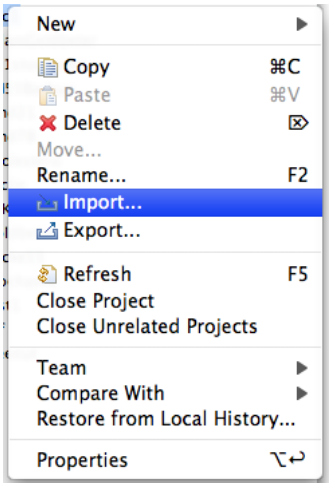



3. Then, choose the projects you want to export, the destination folder, and select options. The default 'Create only selected directories' is normally sufficient. Finally, click **Finish**.

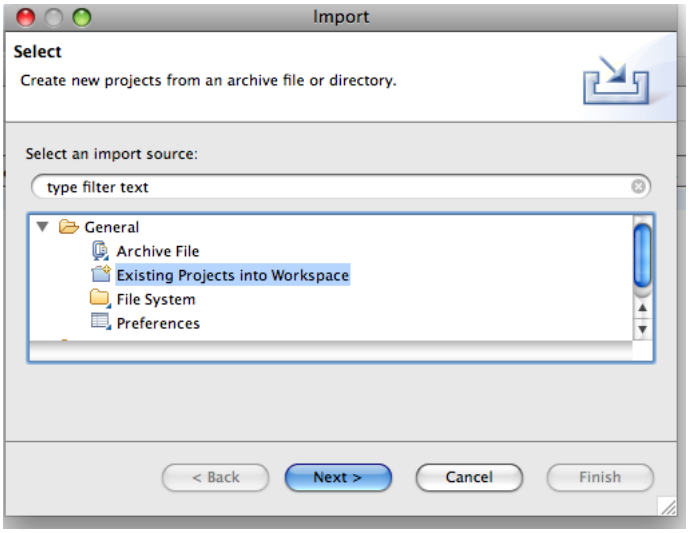



To import a project into the workspace:

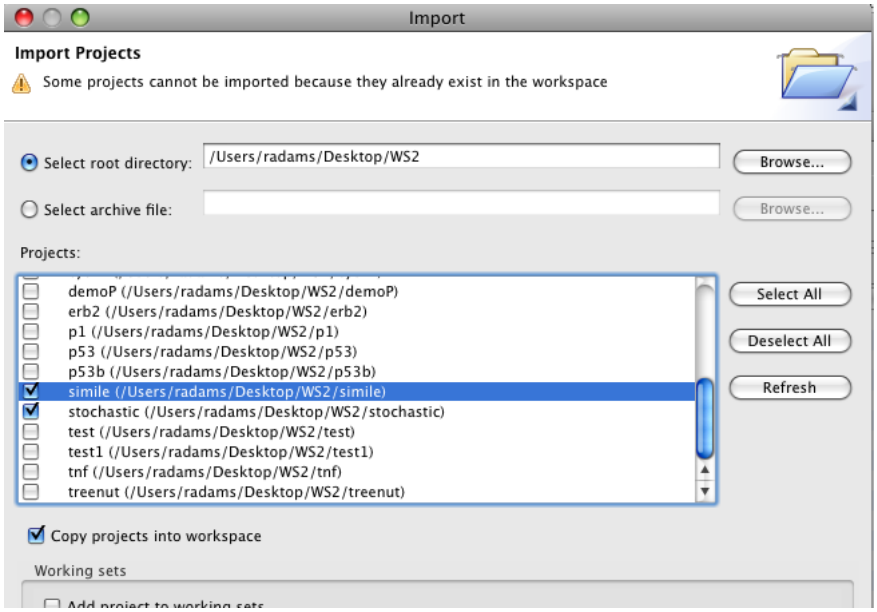
1. Right click anywhere in the System View and choose **Import** from the context menu:




2. Now, choose the 'Import existing projects into workspace' option:




3. Now, browse for the project folder on your filesystem, and ensure it is selected. **You need to select the checkbox to copy resources into your workspace**. Otherwise, the project will continue to exist in its current location and will just appear in the SystemView, without actually being in the workspace folder, which will break the optimization functionality:

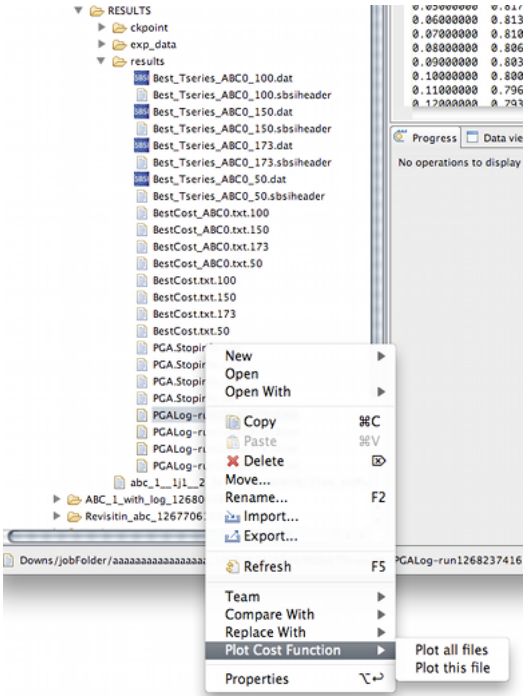



Finally, click finish and the project should appear in the System View.

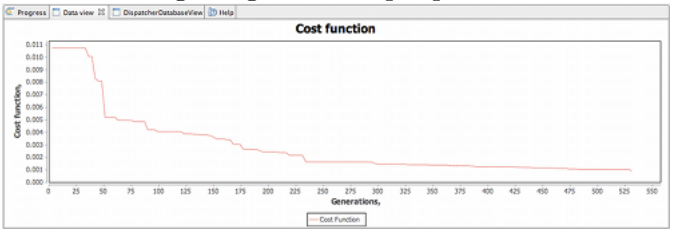
## 5.4 Searching...

Searching functionality is provided by the Search menu:

## 5.5 Visualizing cost function history

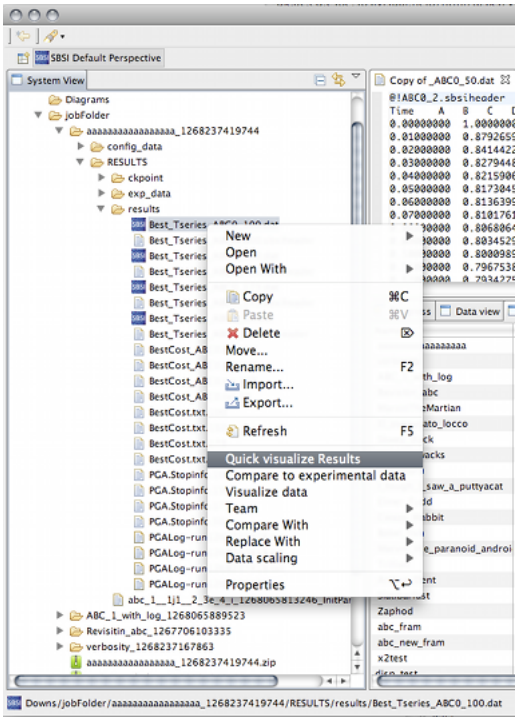



The user has two options in order to plot the cost function values. He can either plot the full span of the simulation, or he can plot a portion of it. By selecting the **Plot all files** action, the user will have a plot with all the values of the plot function that cover all the generations of the simulation. In the case of the **Plot this file** action, the plot will contain a range of the generations, generally those that are included in the selected file. The following image is a sample plot for the whole span of the generations.

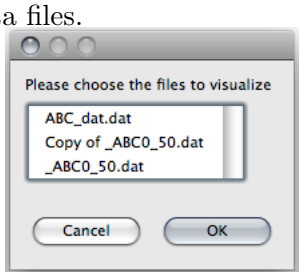



## 5.6 Visualizing optimisation results

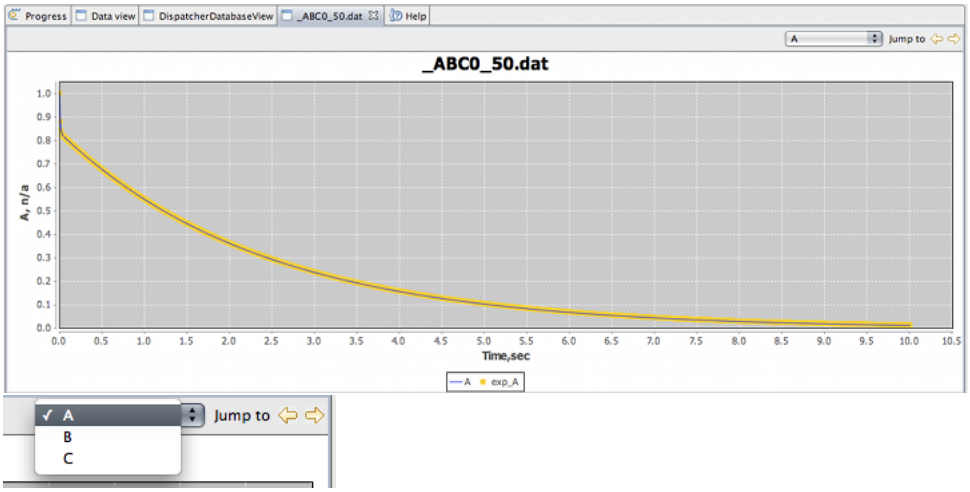
To quick view the result data, you have to navigate inside the RESULTS folder and right click on one of the files that resemble the following: 'Best_Tseries_MODEL_NAME_XXX.dat'.




After selecting the menu option for **Quick visualize results** , the following dialog appears. This dialog presents all the files that exist under the 'exp_data' folder. The user has the option to select one or more of them. By pressing 'OK' the results data will be plotted against the experimental data in individual tabs for any of the experimental data files.




Below we see the plot of one of the tabs. The tab contain all the species that are common between the experimental and the result data. These are being plotted in pairs (experimental against result data). The user has the ability to quickly browse though the species of the model by using the controls on the top right of the view.




## 5.7 Visualizing a time series on a network diagram

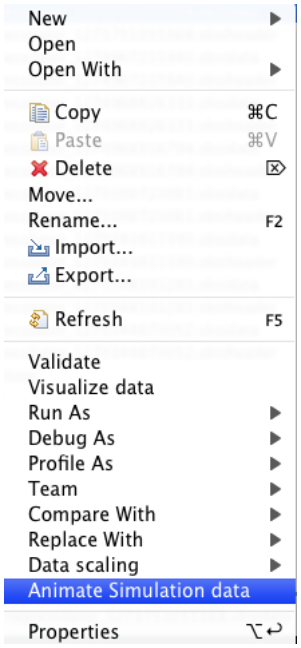
SBSIVisual contains a prototype functionality to display a layout of an SBML file, and overlay time series data on top of it. This functionality is available as an animation.

In order for this to work, you need an SBML model and an experimental data file relevant to your model. To begin with, you need to add a line in the annotation section of the data's header file, which links the data to a model

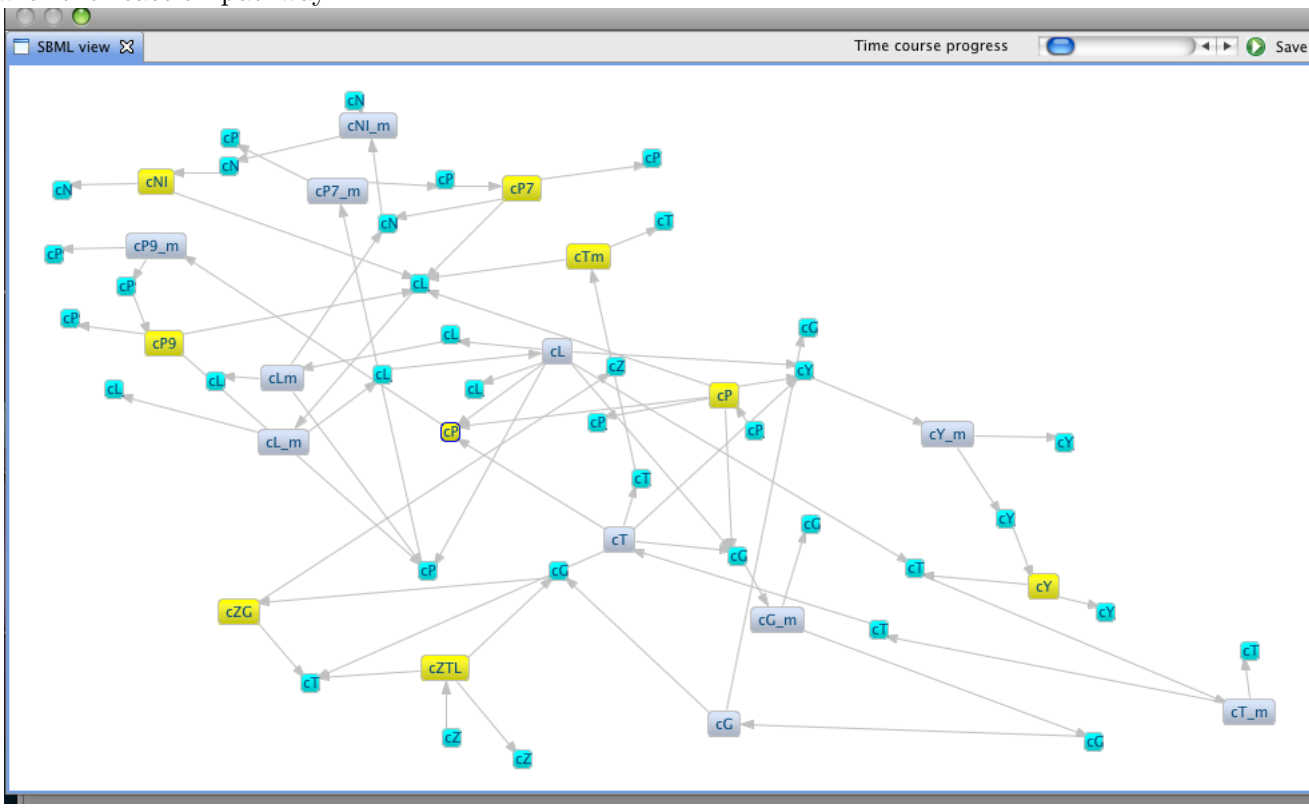
```
*Annotation*
Model ./Models/repressilator.xml
```

The path to the model file can be absolute path on your folder system or a path relative to your data file.

Now, select the data file in the SystemView and select **Animate Simulation data** from the context menu.

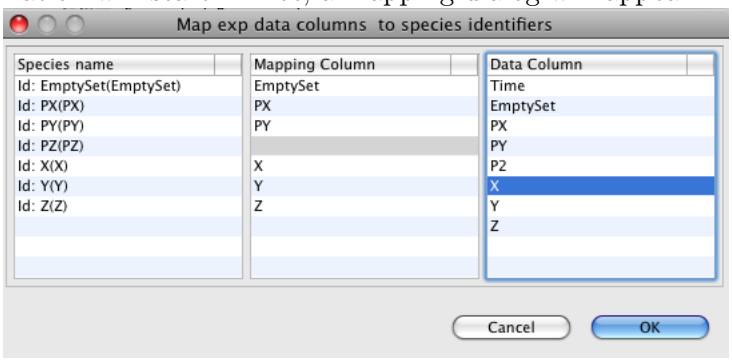



Assuming the path to the model is correct, a view should open showing a simple view of the reaction pathway.




In this view, reactions are small blue nodes and species are larger, pale nodes. You

can rearrange the nodes if the layout is not clear. To animate the nodes, press the play icon on the top-right of the view. If the column names in the data match precisely, the animation will start. If not, a mapping dialog will appear:




Just drag and drop experimental columns from the right-hand side of the dialog into the central section, to match up with the model IDs in the left hand section. Only matching elements will be animated.

Once you're done **click OK** and the animation will start! Nodes resize according to their normalized levels through the time course. E.g., if a node has minimum value 20.0 and maximum value 300, then if, at a given time, the value is 160, the node will appear to be 50% of its maximal size.

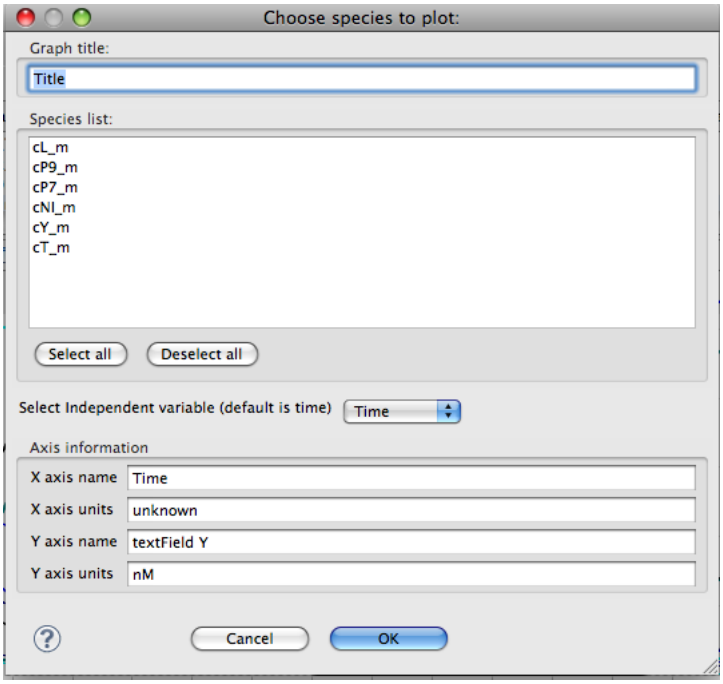
This functionality is also available from the slider, this will proceed frame-by-frame through the animation.

For a non-trivially small model, you'll probably need to edit the layout to make all the nodes visible. Once you've arranged the layout, click on the 'Save animation layout' button and the layout will be preserved. The layout information is saved directly into the SBML model file.

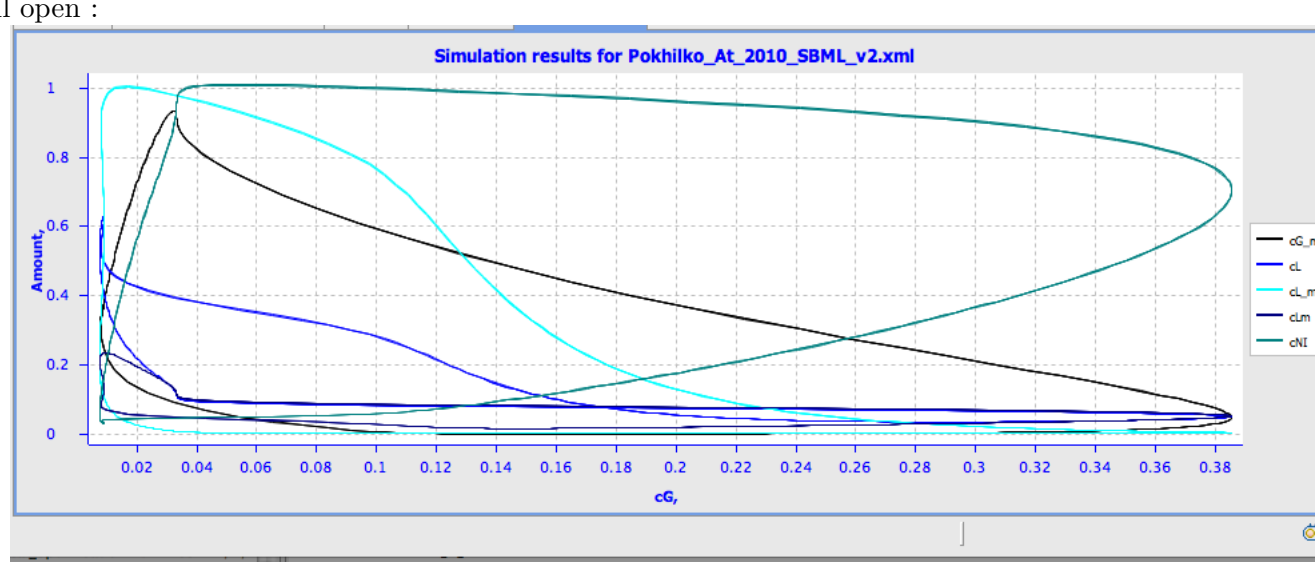
## 5.8 Visualizing time-series data

### 5.8.1 Inbuilt data display

To view time series data, select a file ending in .dat or .sbsidata in the workspace and choose **Visualize data** from the context menu. The following dialog appears:




Using this dialog you can configure the title, axes and units of the graph. A chart will open :




You can use the **Arrow keys** to scroll the axes. If the **Ctrl** key is held down while using the up/down arrow, the graph will be zoomed in/out. Dragging the mouse with the **Shift** key held down will zoom to the selected rectangle. A number of context-menu items are available - restoring the original plot, and showing/hiding the individual data points on the chart. There is also a basic SVG export for data sets viewed via the 'Visualize Data' menu.

For graphs with many data points (tens of thousands), performance is much better

if data points are not displayed.

# Chapter 6

# Reference

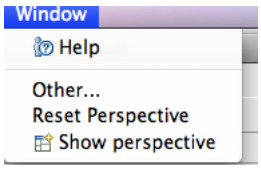
## 6.1 Miscellaneous features

This page provides information on various basic functions provided by SBSIVisual:

- Reset perspective
- Preferences

### 6.1.1 Reset perspective

If you wish to reset the appearance of views and editors in your application, click:

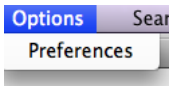



from the Window menu.

This will reset the appearance of the application to its default state. No data is lost during this process, it is purely cosmetic.

### 6.1.2 Preferences

Preferences allow configuration of application-wide settings, and are accessible by:




Preferences can be contributed by any SBSIVisual plugin. By default, SBSIVisual includes the following:

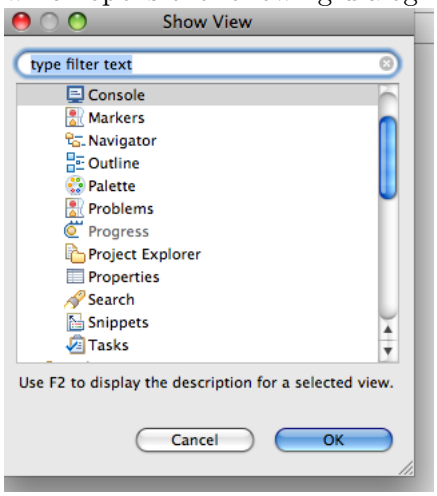
- Preferences to set your SBSI Dispatcher login credentials.
- Preferences to set the extent of SBML validation applied to SBML models.
- Preferences to set the local installation of SBSINumerics, if you have it installed.

## 6.2 Problems view

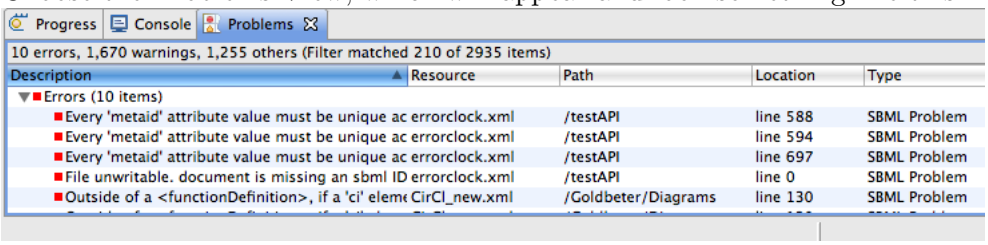
The Problems view shows errors and warnings on the format or content of editable documents. Support is provided for:

- SBML documents.
- SBSI data format data and header files.
- SED-ML simulation archive files.

The Problems view can be opened via : **Window->Other**
which opens the following dialog:

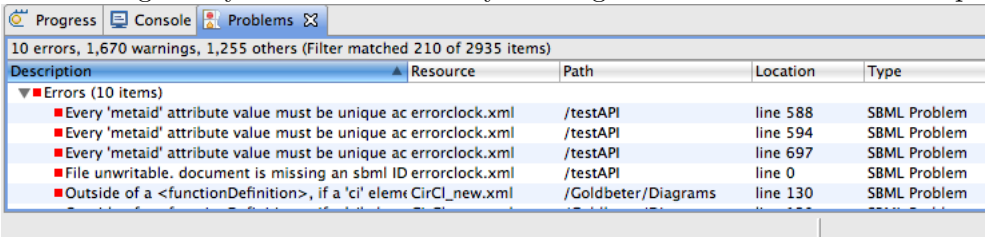



Choose the Problems View, which will appear and look something like this:


Progress
Console
Problems
10 errors, 1,670 warnings, 1,255 others (Filter matched 210 of 2935 items)
Description
Resource
Path
Location
Type
Errors (10 items)
Every 'metaid' attribute value must be unique ac errorclock.xml /testAPI line 588 SBML Problem
Every 'metaid' attribute value must be unique ac errorclock.xml /testAPI line 594 SBML Problem
Every 'metaid' attribute value must be unique ac errorclock.xml /testAPI line 697 SBML Problem
File unwritable. document is missing an sbml ID errorclock.xml /testAPI line 0 SBML Problem
Outside of a <functionDefinition>, if a 'ci' elem CirCl_new.xml /Goldbeter/Diagrams line 130 SBML Problem

Clicking on a line will open an editor. If the system is configured to open an internal editor then the line causing the problem will be selected.

In an editor view, problems are shown by a warning triangle or exclamation marrk in the left margin. If you mouse-over this you will get more information on the problem:

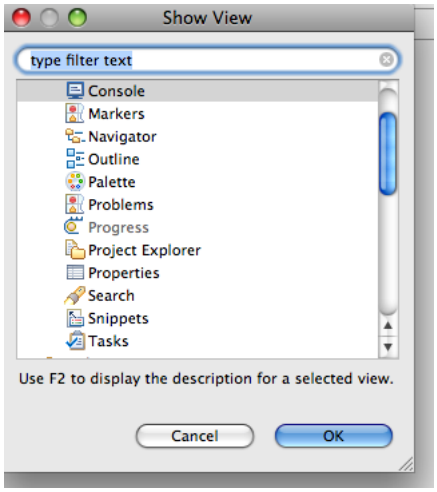

Progress
Console
Problems
10 errors, 1,670 warnings, 1,255 others (Filter matched 210 of 2935 items)
Description
Resource
Path
Location
Type
Errors (10 items)
Every 'metaid' attribute value must be unique ac errorclock.xml /testAPI line 588 SBML Problem
Every 'metaid' attribute value must be unique ac errorclock.xml /testAPI line 594 SBML Problem
Every 'metaid' attribute value must be unique ac errorclock.xml /testAPI line 697 SBML Problem
File unwritable. document is missing an sbml ID errorclock.xml /testAPI line 0 SBML Problem
Outside of a <functionDefinition>, if a 'ci' elem CirCl_new.xml /Goldbeter/Diagrams line 130 SBML Problem

## 6.3 Progress View

The Progress view gives information about long-running processes that are running in the background. The Progress view is enabled at start up or can be opened via :

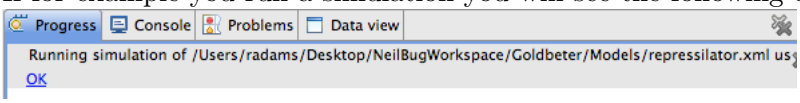
**Window->Other**

which opens the following dialog:




and choose 'Progress'.

If for example you run a simulation you will see the following output:




Clicking on 'OK' will give further information on the outcome of the job.

Very long running jobs will display a progress bar, and provide the option to cancel a long running job.

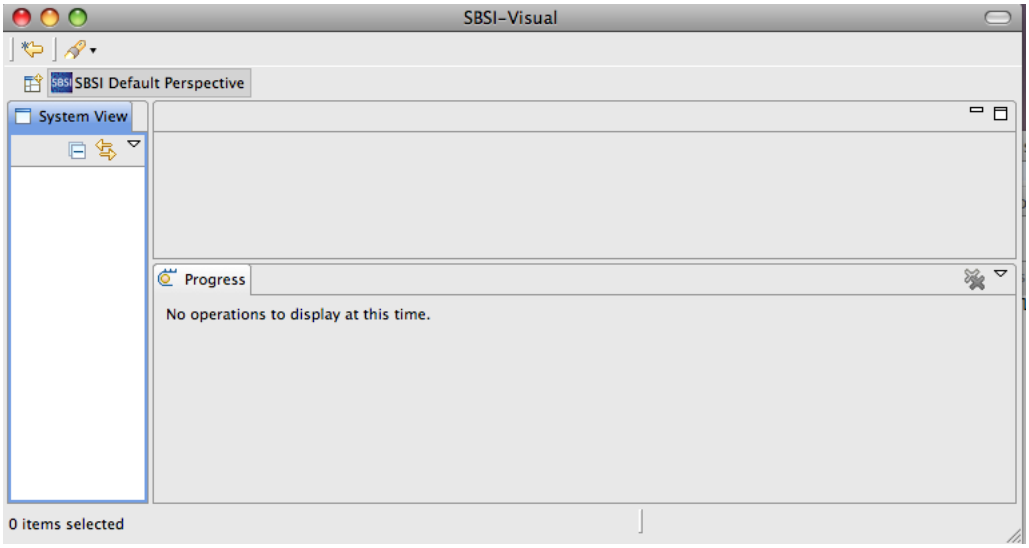
## 6.4 System view

The system view provides a tree view on the underlying file system. Via the context menu you can perform standard move, copy, paste operations, and create new files and folders.

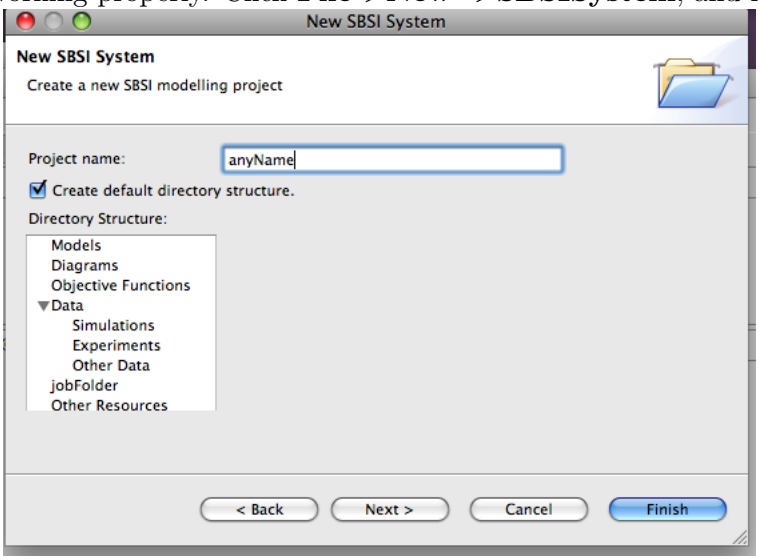
## 6.5 Installation test

Try downloading the application from Sourceforge. Extract the download, go into the application folder and launch the application. (On Windows , this should just be double-clicking the launch icon, on Mac/Linux please read the README file ).

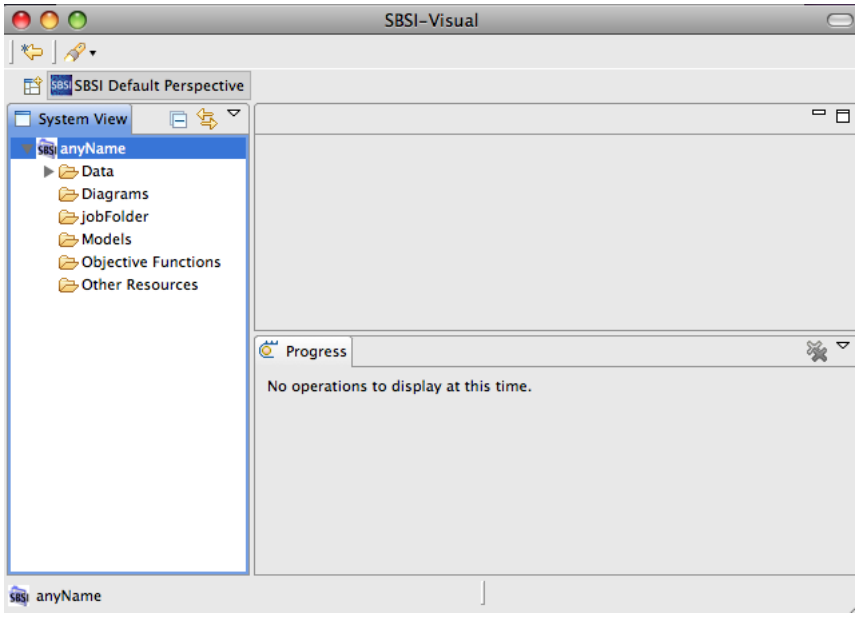
If you've never downloaded SBSIVisual before, you should see a dialog asking you to choose a workspace folder - choose an empty folder and accept defaults, then the application should open. If you have downloaded SBSIVisual before, the application should just open and you should see something like:



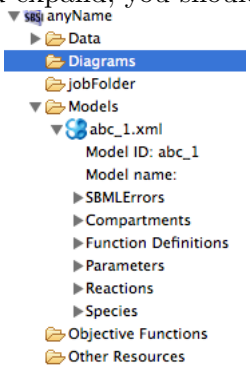

The purpose of this test is to see if the libSBML library included in the application is working properly. Click **File->New ->SBSISystem**, and in the ensuing dialog:




enter any name for the project and click **Finish** . You should see the project created and visible in the 'System View':



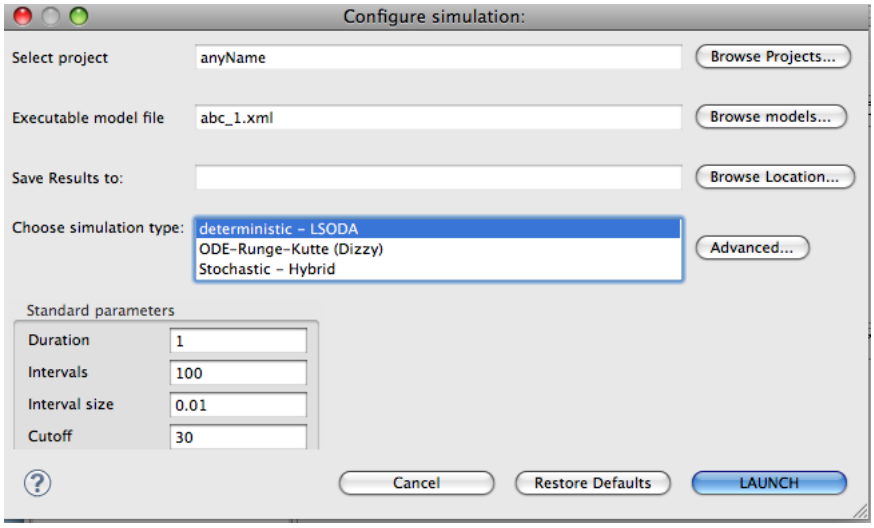

Now, download this simple model (right-click on the link and click 'Download linked file'), then drag the file into any folder in your new project. Click on the file to select and expand, you should see something like this :



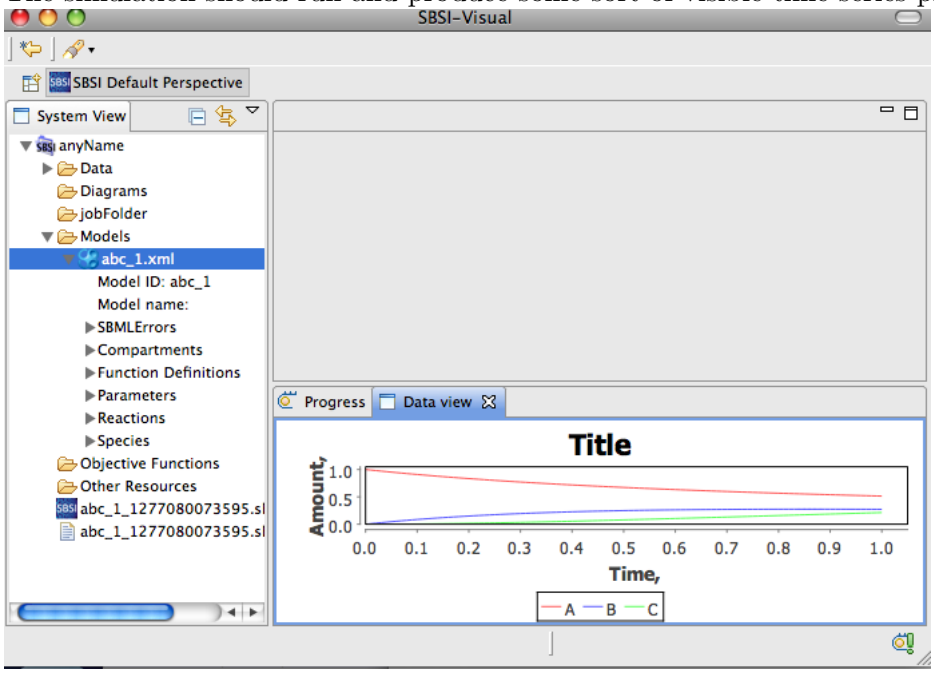

If the file doesn't have the SBML logo decorating it, and doesn't show internal details of the model, then this is a problem!

The aim of this test is to check that the Copasi library for running simulations is working properly.

Select the 'abc_1.xml' model file from the previous step. Now, right-click and choose the 'Simulation' menu item (or Ctrl-R keyboard shortcut) and choose the 'deterministic-LSODA' option, then click 'LAUNCH'. E.g.,




The simulation should run and produce some sort of visible time-series plot. E.g.,




## 6.6 Updating SBSIVisual

### 6.6.1 Updating SBSI

This page provides information on how to update SBSIVisual and install additional functionality:

Once you have downloaded and installed SBSIVisual once, we can provide minor updates via the *update mechanism.* To begin the process, click **Help->Check for**

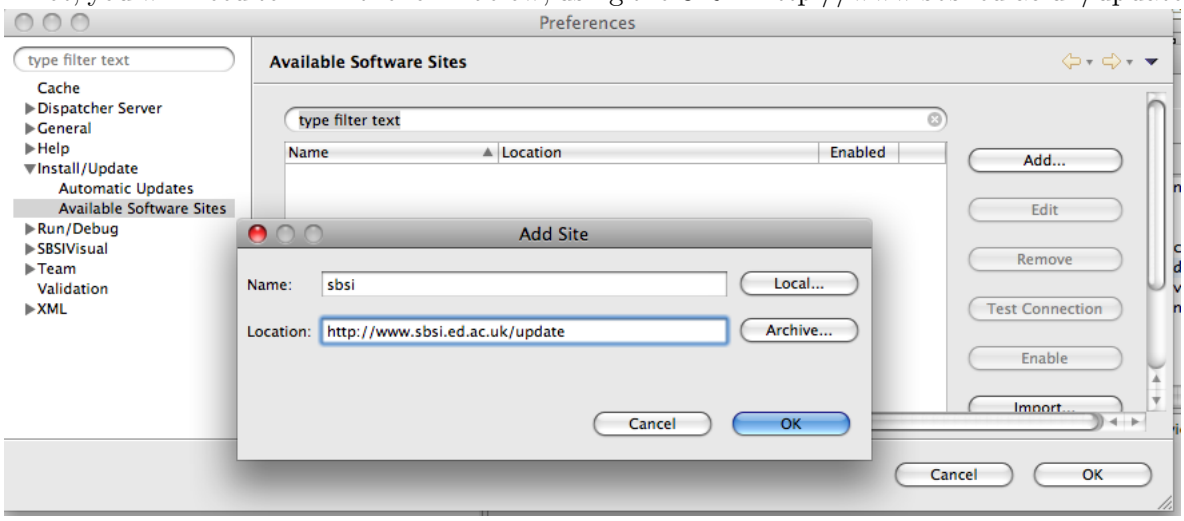
**updates**. If updates are available, choose what you want to update and follow the wizard instructions, setting defaults.

If you want to receive automated notification of updates, you can configure this in the preference pages. Click **Options->preferences** and then expand the **Help/update->automatic update** section.

### 6.6.2 Installing new software

To install new software click **Help->Install New software**. It may be that SBSI is already aware of its own update site - click 'Work with' drop down list to see if a site with: http://www.sbsi.ed.ac.uk/update is mentioned.

If not, you will need to fill in the form below, using the URL "http://www.sbsi.ed.ac.uk/update"




Once you have done this you can **Test Connection** to check all is well.

Now, choose click 'Work with' and choose the SBSI update site. Choose your plugin, and follow the installation procedure. Make sure that the option **Contact all update sites during install...** is checked. This enables SBSI to locate any other required plugins.

You will need to restart the application after updating,so it's best not to do this if you are running a long process. (e.g., a parameter optimization process on your machine).